

NNN	NNN	MM	MM	LL
NNN	NNN	MM	MM	LL
NNN	NNN	MM	MM	LL
NNN	NNN	MMMM	MMMM	LL
NNN	NNN	MMMM	MMMM	LL
NNN	NNN	MMMM	MMMM	LL
NNNNNN	NNN	MM	MM	LL
NNNNNN	NNN	MM	MM	LL
NNNNNN	NNN	MM	MM	LL
NN NNN	NNN	MM	MM	LL
NN NNN	NNN	MM	MM	LL
NN NNN	NNN	MM	MM	LL
NNN	NNNNNN	MM	MM	LL
NNN	NNNNNN	MM	MM	LL
NNN	NNNNNN	MM	MM	LL
NNN	NNN	MM	MM	LL
NNN	NNN	MM	MM	LL
NNN	NNN	MM	MM	LL
NNN	NNN	MM	MM	LLLLLLLLLLLL
NNN	NNN	MM	MM	LLLLLLLLLLLL
NNN	NNN	MM	MM	LLLLLLLLLLLL

FILEID**NMLLISPRM

K 2

NN NN MM MM LL LL I I I I SSSSSSSS PPPPPPPP RRRRRRRR MM MM
NN NN MM MM LL LL I I I I SSSSSSSS PPPPPPPP RRRRRRRR MM MM
NN NN MMMM MMMM LL LL I I SS PP PP RR RR RR MMMM MMMM
NN NN MMMM MMMM LL LL I I SS PP PP RR RR RR MMMM MMMM
NNNN NN MM MM LL LL I I SS PP PP RR RR RR MM MM MM
NNNN NN MM MM LL LL I I SS PP PP RR RR RR MM MM MM
NN NN NN MM MM LL LL I I SSSSSS PPPPPPPP RRRRRRRR MM MM
NN NN NN MM MM LL LL I I SSSSSS PPPPPPPP RRRRRRRR MM MM
NN NNNN MM MM LL LL I I SS PP RR RR RR MM MM
NN NNNN MM MM LL LL I I SS PP RR RR RR MM MM
NN NN MM MM LL LL I I SS PP RR RR RR MM MM
NN NN MM MM LL LL I I SS PP RR RR RR MM MM
NN NN MM MM LLLLLLLL LLLLLLLL I I I I SSSSSSSS PP RR RR MM MM
NN NN MM MM LLLLLLLL LLLLLLLL I I I I SSSSSSSS PP RR RR MM MM
LL I I I I SSSSSSSS
LL I I I I SSSSSSSS
LL SS SS
LLLLLLL LLLL LLLL SSSSSSSS
LLLLLLL LLLL LLLL SSSSSSSS

NMI
VO
:
::

```
0001 0 XTITLE 'NML special parameter handling routines'
0002 0 MODULE NMLLISPRM {
0003 0   LANGUAGE (BLISS32),
0004 0   ADDRESSING_MODE (NONEXTERNAL=GENERAL),
0005 0   ADDRESSING_MODE (EXTERNAL=GENERAL),
0006 0   IDENT = 'V04-000'
0007 0   )
0008 1 BEGIN
0009 1 ****
0010 1 *
0011 1 *
0012 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0013 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0014 1 * ALL RIGHTS RESERVED.
0015 1 *
0016 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0017 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0018 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0019 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0020 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0021 1 * TRANSFERRED.
0022 1 *
0023 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0024 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0025 1 * CORPORATION.
0026 1 *
0027 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0028 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0029 1 *
0030 1 *
0031 1 ****
0032 1 *
0033 1 *
0034 1 ++
0035 1 FACILITY: DECnet-VAX V2.0 Network Management Listener
0036 1
0037 1 ABSTRACT:
0038 1
0039 1 This module contains action routines to handle changing and
0040 1 displaying of permanent data base entity parameters.
0041 1
0042 1 ENVIRONMENT: VAX/VMS Operating System
0043 1
0044 1 AUTHOR: Distributed Systems Software Engineering
0045 1
0046 1 CREATION DATE: 23-JAN-1980
0047 1
0048 1 MODIFIED BY:
0049 1
0050 1 V03-008 MKP0009 Kathy Perko 2-Aug-1984
0051 1 Fix DEFINE EXEC ADDR n so that, if n doesn't include an area
0052 1 number, area 1 is used.
0053 1
0054 1 V03-007 MKP0008 Kathy Perko 20-April-1984
0055 1 Fix DEF NODE nnn ADDR yyy so that, if the address is a duplicate
0056 1 of the executor's, the error message indicates "executor"
0057 1 instead of "remote node".
```

58 0058 1
59 0059 1 V03-006 MKP0007 Kathy Perko 18-April-1984
60 0060 1 Fix DEF EXEC NAME or ADDRESS so that exec id globals
61 0061 1 are updated.
62 0062 1
63 0063 1 V03-005 MKP0006 Kathy Perko 29-Jan-1984
64 0064 1 If NCP is a V3.0.0, mask area in node numbers.
65 0065 1
66 0066 1 V03-004 MKP0005 Kathy Perko 4-Aug-1983
67 0067 1 Change routines to manipulate permanent database record
68 0068 1 fields to be transparent to ISAM keys at the beginning of
69 0069 1 the records. Also, redo checking on node ids for the new
70 0070 1 node database format.
71 0071 1
72 0072 1 V03-003 MKP0004 Kathy Perko 29-July-1983
73 0073 1 Redo NML\$LISTNODEID routine to return only the node id if
74 0074 1 the PSTs datatype is NMASM_PTY_CM1.
75 0075 1
76 0076 1 V03-002 MKP0003 Kathy Perko 13-July-1982
77 0077 1 Fix NML\$LISTPARAM to add parameter lengths correctly.
78 0078 1 Fix list routines for channels and set passwords.
79 0079 1
80 0080 1 V03-001 MKP0002 Kathy Perko 16-June-1982
81 0081 1 Add new list routines for range and circuit owner paramters.
82 0082 1
83 0083 1 V02-001 MKP0001 Kathy Perko 2-April-1982
84 0084 1 Add changes for X-25 Protocol Networks and DTE, and
85 0085 1 for X-25 Server Modules.
86 0086 1
87 0087 1 V02-001 MKP001 Kathy Perko 24-July-1981
88 0088 1 Delete NML call to map VMS line to DNA line name and
89 0089 1 vice versa.
90 0090 1
91 0091 1 --

```
93      0092 1 %SBTTL 'Declarations'  
94      0093 1  
95      0094 1  
96      0095 1 ! TABLE OF CONTENTS:  
97      0096 1 !  
98      0097 1 !  
99      0098 1 FORWARD ROUTINE  
100     0099 1 NMLSLISNMLVER,  
101     0100 1 NMLSLISLONAM,  
102     0101 1 NMLSLISNODEID,  
103     0102 1 NMLSLISPARAM,  
104     0103 1 NMLSLISPASSWORD,  
105     0104 1 NMLSLISPSET,  
106     0105 1 NMLSLISRANGE,  
107     0106 1 NMLSLISOWNER,  
108     0107 1 NMLSLDEFPARAM,  
109     0108 1 NMLSLDEFLINLY,  
110     0109 1 NMLSLDEFLINTRI,  
111     0110 1 NMLSLDEF_NODE_ADDR,  
112     0111 1 NMLSLDEF_EXEC_ID,  
113     0112 1 NMLSLDEFNDNLI,  
114     0113 1 NMLSLDEFOBJNUM,  
115     0114 1 NMLSPURPARAM,  
116     0115 1 NMLSPURNODNNA;  
117     0116 1  
118     0117 1  
119     0118 1 ! INCLUDE FILES:  
120     0119 1  
121     0120 1 !  
122     0121 1  
123     0122 1 LIBRARY 'LIB$:NMLLIB.L32';  
124     0123 1 LIBRARY 'SHRLIB$:NMALIBRY.L32';  
125     0124 1 LIBRARY 'SYSSLIBRARY:STARLET.L32';  
126     0125 1  
127     0126 1 ! OWN STORAGE:  
128     0127 1  
129     0128 1 !  
130     0129 1  
131     0130 1 !  
132     0131 1 Parameter buffer and descriptor for use in handling volatile data base  
133     0132 1 data.  
134     0133 1 !  
135     0134 1 OWN  
136     0135 1     nmlSt_prmbuffer : VECTOR [256, BYTE];  
137     0136 1 BIND  
138     0137 1     nmlSq_prmdsc = UPLIT (256, nmlSt_prmbuffer) : DESCRIPTOR;  
139     0138 1 !  
140     0139 1 Entity buffer and descriptor.  
141     0140 1 !  
142     0141 1 OWN  
143     0142 1     nmlSt_entbuffer : BBLOCK [nmlSk_entbuflen],  
144     0143 1     nmlSq_entbfdsc : VECTOR [2];  
145     0144 1 !  
146     0145 1 ! EXTERNAL REFERENCES:  
147     0146 1 !  
148     0147 1 !  
149     0148 1 !
```

```
: 150      0149 1 $NML_EXTDEF;  
151      0150 1  
152      0151 1 EXTERNAL LITERAL  
153      0152 1     nml$_recbfovf,  
154      0153 1     nml$_recdelet;  
155      0154 1  
156      0155 1 EXTERNAL  
157      0156 1     nml$gw_perm_exec_addr : BBLOCK [2],  
158      0157 1     nml$gb_ncp_version,  
159      0158 1     nml$gq_perm_exec_name_dsc : VECTOR [2],  
160      0159 1     nml$gq_proprvmsk : BBLOCK [8];  
161      0160 1  
162      0161 1 EXTERNAL ROUTINE  
163      0162 1     nma$deletefld,  
164      0163 1     nma$insertfld,  
165      0164 1     nma$matchrec,  
166      0165 1     nma$searchfld,  
167      0166 1     nml$addmsgprm,  
168      0167 1     nml$bld_reply,  
169      0168 1     nml$delete_node_rec,  
170      0169 1     nml$getexeaddr,  
171      0170 1     nml$getnodnam,  
172      0171 1     nml$getrecowner,  
173      0172 1     nml$read_loopnode,  
174      0173 1     nml$readrecord,  
175      0174 1     nml$send;  
176      0175 1
```

```
178 0176 1 %SBTTL 'NML$LISNMLVER Get NML version number'
179 0177 1 GLOBAL ROUTINE NML$LISNMLVER (SEM_TABLE, BUFDSC, MSGSIZE, DUMDSC) =
180 0178 1
181 0179 1 !++
182 0180 1 ! FUNCTIONAL DESCRIPTION:
183 0181 1
184 0182 1 This routine moves the network management version number into
185 0183 1 the output message as a coded multiple parameter.
186 0184 1
187 0185 1 ! FORMAL PARAMETERS:
188 0186 1
189 0187 1 SEM_TABLE Parameter semantic table entry address.
190 0188 1 BUFDSC Output message buffer descriptor.
191 0189 1 MSGSIZE Address of current output message size.
192 0190 1 DUMDSC Not used.
193 0191 1
194 0192 1 ! IMPLICIT INPUTS:
195 0193 1
196 0194 1 It is assumed that the permanent data base file is already open.
197 0195 1
198 0196 1 ! IMPLICIT OUTPUTS:
199 0197 1
200 0198 1 Parameter is added to output message buffer.
201 0199 1
202 0200 1 ! ROUTINE VALUE:
203 0201 1 ! COMPLETION CODES:
204 0202 1
205 0203 1 Always returns success (NML$_STS_SUC).
206 0204 1
207 0205 1 ! SIDE EFFECTS:
208 0206 1
209 0207 1 ! NONE
210 0208 1
211 0209 1 !--
212 0210 1
213 0211 2 ! BEGIN
214 0212 2
215 0213 2 ! MAP
216 0214 2 SEM_TABLE : REF BBLOCK;
217 0215 2
218 0216 2 ! LOCAL
219 0217 2 BUFFER : VECTOR [6, BYTE],
220 0218 2 PTR;
221 0219 2
222 0220 2 PTR = CHSPTR (BUFFER); ! Get pointer to output buffer
223 0221 2
224 0222 2 ! Add version numbers preceded by data type.
225 0223 2
226 0224 2 ! CHSWCHAR_A (1, PTR);
227 0225 2 CHSWCHAR_A (NML$K_VERSION, PTR);
228 0226 2 CHSWCHAR_A (1, PTR);
229 0227 2 CHSWCHAR_A (NML$K_DEC_ECO, PTR);
230 0228 2 CHSWCHAR_A (1, PTR);
231 0229 2 CHSWCHAR_A (NML$K_USER_ECO, PTR);
232 0230 2
233 0231 2
234 0232 2 !
```

```

235 0233 2 ! Add coded multiple version parameter to message.
236 0234 2 !
237 0235 2 NML$ADDMMSGPRM (.BUFDSC,
238 0236 2 .MSGSIZE,
239 0237 2 .SEM_TABLE [PST$W_DATAID],
240 0238 2 .SEM_TABLE [PST$B_DATATYPE] OR 3,
241 0239 2 6,
242 0240 2 BUFFER);
243 0241 2
244 0242 2 RETURN NMLS_STS_SUC
245 0243 2
246 0244 1 END: ! End of NML$LISNMLVER

.TITLE NML$LISPRM NML special parameter handling routines
.IDENT \V04-000\
.PSECT SPLITS,NOWRT,NOEXE,2
00000100 00000 P.AAA: .LONG 256
00000000 00004 .ADDRESS NML$T_PRMBUFFER
.PSECT SOWNS,NOEXE,2
00000 NML$T_PRMBUFFER:
00100 NML$T_ENTBUFFER: .BLKB 256
00140 NML$Q_ENTBFDSC: .BLKB 64
00140 NML$Q_ENTBFDSC: .BLKB 8

NML$Q_PRMDSC= P.AAA
.EXTRN NML$GB_EVTSRCTYP
.EXTRN NML$GQ_EVTSRCDS
.EXTRN NML$GW_EVTCLASS
.EXTRN NML$GB_EVTMSKTYP
.EXTRN NML$GQ_EVTMSKDSC
.EXTRN NML$GW_EVTSNKADR
.EXTRN NML$GW_ACP_CHAN
.EXTRN NML$GL_LOGMASK, NML$GQ_ENTSTRDSC
.EXTRN NML$AB_QIOBUFFER
.EXTRN NML$GQ_QIOBFDSC
.EXTRN NML$AB_EXEBUFFER
.EXTRN NML$GL_EXEDATPTR
.EXTRN NML$GQ_EXEDATDSC
.EXTRN NML$GQ_EXEBFDSC
.EXTRN NML$AB_RCVBUFFER
.EXTRN NML$GQ_RCVFDSC
.EXTRN NML$AB_SNDBUFFER
.EXTRN NML$GQ_SNDBFDSC
.EXTRN NML$GL_RCVDATLEN
.EXTRN NML$AB_CPTABLE, NML$AB_MSGBLOCK
.EXTRN NML$AB_ENTITY_ID
.EXTRN NML$AB_QUALIFIER_ID
.EXTRN NML$AB_ENTITYDATA
.EXTRN NML$AB_NML_NMV, NML$AB_PRMSEM

```

NMLSL ISPRM
V04-000

NML special parameter handling routines
NML\$LI\$NMLVER Get NML version number

5

16-Sep-1984 00:16:56
14-Sep-1984 12:50:09

16-Sep-1984 00:16:56 VAX-11 BLiss-32 V4.0-742
14-Sep-1984 12:50:09 [NML.SRC]NML LISPRM-B32:1

Page 7
(3)

NMI
VOI

• EXTRN NML\$AB_RECVBUF, NML\$AL_ENTINFTAB
• EXTRN NML\$AL_PERMINFTAB
• EXTRN NML\$AW_PRM DES, NML\$GB_CMD_VER
• EXTRN NML\$GB_ENTITY_CODE
• EXTRN NML\$GB_ENTITY_FORMAT
• EXTRN NML\$GL_QUALIFIER_PST
• EXTRN NML\$GB_QUALIFIER_FORMAT
• EXTRN NML\$GB_FUNCTION
• EXTRN NML\$GB_INFO, NML\$GB_OPTIONS
• EXTRN NML\$GL_PRMCODE, NML\$GL_PRS_FLGS
• EXTRN NML\$GL_NML ENTITY
• EXTRN NML\$GQ_NETRNAME DSC
• EXTRN NML\$GQ_RECVFDSC
• EXTRN NML\$GW_PRMDESCNT
• EXTRN NML\$ RECBFOVF, NML\$ RECDELET
• EXTRN NML\$GW_PERM EXEC ADDR
• EXTRN NML\$GB_NCP VERSION
• EXTRN NML\$GQ_PERM EXEC_NAME_DSC
• EXTRN NML\$GQ_PROPRTVMSK
• EXTRN NMA\$DELETEFLD, NMA\$INSERTFLD
• EXTRN NMA\$MATCHREC, NMA\$SEARCHFLD
• EXTRN NML\$ADDMSGPRM, NML\$BLD_REPLY
• EXTRN NML\$DELETE NODE REC
• EXTRN NML\$GETEXEADR, NML\$GETNODNAME
• EXTRN NML\$GETRECOwner
• EXTRN NML\$READ_LOOPNODE
• EXTRN NML\$READRECORD, NML\$SEND

```
.PSECT SCODES,NOWRT,2
.ENTRY NMLSLISNMLVER, Save nothing
SUBL2 #8, SP
MOVAB BUFFER, PTR
MOVL #66561, (PTR) +
MOVW #1, (PTR) +
PUSHL SP
PUSHL #6
MOVL SEM TABLE, R0
MOVZBL 3(R0), R1
BISL3 #3, R1, -(SP)
MOVZWL (R0), -(SP)
MOVO BUFDESC, -(SP)
CALLS #6, NMLSADDMSGPRM
MOVL #1, R0
RET
```

		0000	00000
5E		08	C2 00002
50		6E	9E 00005
80	00010401	8F	00 00008
80		01	B0 0000F
		5E	D' 00012
		06	5D 00014
50	04	AC	D0 00016
51	03	A0	9A 0001A
51		03	C9 0001E
7E		60	3C 00022
7E	08	AC	7D 00025
00000000G	00	06	FB 00029
	50	01	D0 00030
		06	00033

; Routine Size: 52 bytes, Routine Base: \$CODES + 0000

```
248 0245 1 %SBTTL 'NML$LISLOONAM Get loop node name'  
249 0246 1 GLOBAL ROUTINE NML$LISLOONAM (SEM_LIST, BUFDSC, MSGSIZE, DATDSC)=  
250 0247 1  
251 0248 1 !++  
252 0249 1 : FUNCTIONAL DESCRIPTION:  
253 0250 1 :  
254 0251 1 : This routine returns the loopback node name for a line.  
255 0252 1 :  
256 0253 1 : FORMAL PARAMETERS:  
257 0254 1 :  
258 0255 1 : SEM_LIST Parameter semantic table entry address.  
259 0256 1 : BUFDSC Output message buffer descriptor address.  
260 0257 1 : MSGSIZE Address of current output message size.  
261 0258 1 : DATDSC Data buffer descriptor address.  
262 0259 1 :  
263 0260 1 : IMPLICIT INPUTS:  
264 0261 1 :  
265 0262 1 : It is assumed that the permanent data base file is already open.  
266 0263 1 :  
267 0264 1 : ROUTINE VALUE:  
268 0265 1 : COMPLETION CODES:  
269 0266 1 :  
270 0267 1 : Always returns success (NMLS_STS_SUC).  
271 0268 1 :  
272 0269 1 : SIDE EFFECTS:  
273 0270 1 :  
274 0271 1 : NONE  
275 0272 1 :  
276 0273 1 :--  
277 0274 1 :  
278 0275 2 BEGIN  
279 0276 2  
280 0277 2 MAP  
281 0278 2 sem_list : REF BBLOCK;  
282 0279 2  
283 0280 2 LOCAL  
284 0281 2 circuit_dsc : VECTOR [2].  
285 0282 2 node_dsc : VECTOR [2].  
286 0283 2 node_rec_buf: BBLOCK [nml$k_recflen], ! Buffer for node data  
287 0284 2 node_rec_dsc: VECTOR [2]. ! Descriptor of node data buffer  
288 0285 2 node_rec_data:VECTOR [2]. ! Descriptor of data in node  
289 0286 2 data buffer.  
290 0287 2 status:  
291 0288 2  
292 0289 2  
293 0290 2 :  
294 0291 2 : Get the circuit ID from the circuit's permanent database record.  
295 0292 2 : If this fails, it's a bug.  
296 0293 2 :  
297 0294 2 circuit_dsc [0] = 0;  
298 0295 2 circuit_dsc [1] = 0;  
299 0296 2 IF NOT nma$searchfld (.datdsc,  
300 0297 2 nml$c_key_cir,  
301 0298 2 circuit_dsc [0],  
302 0299 2 circuit_dsc [1]) THEN  
303 0300 2 RETURN nml$sts_mpr;  
304 0301 2 node_rec_dsc [0] = nml$k_recflen;
```

```

305 0302 2 node_rec_dsc [1] = node_rec_buf;
306 0303 2 node_rec_data [1] = node_rec_buf;
307 0304 1
308 0305 2 : Call routine to read through the known loopnodes in the node permanent
309 0306 2 database, looking for loopnode on the circuit being listed.
310 0307 2
311 0308 2 IF nml$read_loopnode (circuit_dsc,
312 0309 2           node_rec_dsc,
313 0310 2           node_rec_data) THEN
314 0311 1 BEGIN
315 0312 1   node_dsc [0]= 0;
316 0313 1   node_dsc [1] = 0;
317 0314 1   IF nma$searchfld (node_rec_data,
318 0315 1           nma$c_pcno_nna,
319 0316 1           node_dsc [0],
320 0317 1           node_dsc [1]) THEN
321 0318 1     nml$addmsgprm (.bufdsc,
322 0319 1           .msgsize,
323 0320 1           .sem_list [pst$w_dataid],
324 0321 1           .sem_list [pst$b_datatype],
325 0322 1           .node_dsc [0],
326 0323 1           .node_dsc [1]);
327 0324 1 END;
328 0325 2 RETURN nml$_sts_suc
329 0326 1 END;

```

! End of NML\$LISLOONAM

				0004 00000	.ENTRY	NML\$LISLOONAM, Save R2	0246
52	00000000G	00	9E	00002	MOVAB	NMASSEARCHFLD, R2	
SE	FBE0	CE	9E	00009	MOVAB	-1056(SP), SP	
		F8	AD	7C 0000E	CLRQ	CIRCUIT_DSC	0294
		FC	AD	9F 00011	PUSHAB	CIRCUIT_DSC+4	0299
		F8	AD	9F 00014	PUSHAB	CIRCUIT_DSC	0298
7E		04	CE	00017	MNEGL	#4, -(SP)	0296
		10	AC	DD 0001A	PUSHL	DATA_DSC	
62		04	FB	0001D	CALLS	#4, NMASSEARCHFLD	
04		50	E8	00020	BLBS	R0, 1\$	
50		0A	CE	00023	MNEGL	#10, R0	0300
				04 00026	RET		
08	AE	0400	8F	3C 00027 1\$:	MOVZWL	#1024, NODE REC DSC	0301
0C	AE	10	AE	9E 0002D	MOVAB	NODE_REC_BUF, NODE_REC_DSC+4	0302
04	AE	10	AE	9E 00032	MOVAB	NODE_REC_BUF, NODE_REC_DATA+4	0303
		5E	DD	00037	PUSHL	SP	0308
		OC	AE	9F 00039	PUSHAB	NODE REC DSC	
		F8	AD	9F 0003C	PUSHAB	CIRCUIT_DSC	
00000000G	00	03	FB	0003F	CALLS	#3, NML\$READ_LOOPNODE	
	31	50	E9	00046	BLBC	R0, 2\$	
		F0	AD	7C 00049	CLRQ	NODE_DSC	0312
		F4	AD	9F 0004C	PUSHAB	NODE_DSC+4	0317
		F0	AD	9F 0004F	PUSHAB	NODE_DSC	0316
7E	01F4	8F	3C	00052	MOVZWL	#500, -(SP)	0314
		0C	AE	9F 00057	PUSHAB	NODE REC DATA	
62		04	FB	0005A	CALLS	#4, NMASSEARCHFLD	
1A		50	E9	0005D	BLBC	R0, 2\$	

NML\$LISPRM
V04-000

NML special parameter handling routines
NML\$LISLOONAM Get loop node name

H 3
16-Sep-1984 00:16:56
14-Sep-1984 12:50:09

VAX-11 Bliss-32 V4.0-742
[NML.SRC]NMLLISPRM.B32;1

Page 10
(4)

7E	F0	AD	7D	00060	MOVQ	NODE DSC, -(SP)	: 0322
50	04	AC	00	00064	MOVL	SEM [IST, R0	: 0321
7E	03	A0	0A	00068	MOVZBL	3(R0), -(SP)	: 0320
7E	60	3C	0006C	MOVZWL	(R0), -(SP)	: 0318	
7E	08	AC	7D	0006F	MOVQ	BUFDSC, -(SP)	: 0325
00000000G	00	06	FB	00073	CALLS	#6, NML\$ADDMSGPRM	: 0326
50	01	00	0007A	28:	MOVL	#1, R0	
	04	0007D			RET		

; Routine Size: 126 bytes, Routine Base: SCODES + 0034

```
331 0327 1 %SBTTL 'NML$LISNODEID Get host node id'  
332 0328 1 GLOBAL ROUTINE NML$LISNODEID (SEM_LIST, BUFDSC, MSGSIZE, DATDSC)=  
333 0329 1 !++  
334 0330 1 FUNCTIONAL DESCRIPTION:  
335 0331 1 This routine gets the host node identification string.  
336 0332 1 FORMAL PARAMETERS:  
337 0333 1 SEM LIST Parameter semantic table entry address.  
338 0334 1 BUFDSC Output message buffer descriptor address.  
339 0335 1 MSGSIZE Address of current output message size.  
340 0336 1 DATDSC Data buffer descriptor address.  
341 0337 1 IMPLICIT INPUTS:  
342 0338 1 It is assumed that the permanent data base file is already open.  
343 0339 1 IMPLICIT OUTPUTS:  
344 0340 1 NONE  
345 0341 1 ROUTINE VALUE:  
346 0342 1 COMPLETION CODES:  
347 0343 1 Always returns success (NML$_STS_SUC).  
348 0344 1 SIDE EFFECTS:  
349 0345 1 NONE  
350 0346 1 --  
351 0347 2 BEGIN  
352 0348 2 MAP  
353 0349 2 sem_list : REF BBLOCK;  
354 0350 2 OWN  
355 0351 2 tmpbuffer : BBLOCK [6];  
356 0352 2 BIND  
357 0353 2 tmpdsc = UPLIT (6, tmpbuffer) : DESCRIPTOR;  
358 0354 2 LOCAL  
359 0355 2 cm_count,  
360 0356 2 fldadr,  
361 0357 2 fldsize,  
362 0358 2 length,  
363 0359 2 namdsc : DESCRIPTOR,  
364 0360 2 hostadr : WORD,  
365 0361 2 ptr  
366 0362 2 reslen;  
367 0363 2 fldadr = 0;  
368 0364 2 IF NOT nma$searchfld (.datdsc,
```

```
388 0384 2 .sem_list [pst$w_dataid],  
389 0385 2 fldsize,  
390 0386 2 fldadr) THEN  
391 0387 2 RETURN nml$sts_pty;  
392 0389 2 ptr = nml$st_prmbuffer;  
393 0390 2 : Get the maximum number of fields in the coded multiple: 1 (node address  
394 0391 2 only) or 2 (node address and node name).  
395 0393 2 : cm_count = .sem_list [pst$b_datatype] AND NOT nma$m_pty_cmu;  
396 0394 2  
397 0395 2 hostadr = .(.fldadr)<0,16>;  
398 0396 2 : Add node address field.  
399 0399 2 CHSWCHAR_A (2, ptr);  
400 0400 2 : If the NCP I'm talking to is speaking NICE V3.0.0 or less, clear the  
401 0402 2 area number from node numbers in the executor's area.  
402 0403 2  
403 0404 2 IF CHSRCHAR (nml$gb_ncp_version) LEQ 3 THEN  
404 0405 3 BEGIN  
405 0406 3 MAP  
406 0407 3 hostadr : BBLOCK [2];  
407 0408 3  
408 0409 3 IF .hostadr [nma$v_area] EQL .nml$gw_perm_exec_addr [nma$v_area] THEN  
409 0410 3 hostadr [nma$v_area] = 0;  
410 0411 3  
411 0412 2 END;  
412 0413 2  
413 0414 2 ptr = CHSMOVE (2, hostadr, .ptr);  
414 0415 2 IF .cm_count EQL 2 THEN  
415 0416 3 BEGIN  
416 0417 3 nml$getnodnam (.hostadr, tmpdsc, reslen);  
417 0418 3 namdsc [dsc$w_length] = .reslen;  
418 0419 3 namdsc [dsc$w_pointer] = tmpbuffer;  
419 0420 3 : Add node name field if the length is not zero.  
420 0421 3  
421 0422 3 IF .namdsc [dsc$w_length] NEQU 0 THEN  
422 0423 4 BEGIN  
423 0424 4 CHSWCHAR_A (nma$m_pty_asc, ptr);  
424 0425 4 CHSWCHAR_A (.namdsc [dsc$w_length], ptr);  
425 0426 4 ptr = CHSMOVE (.namdsc [dsc$w_length],  
426 0427 4 .namdsc [dsc$w_pointer],  
427 0428 4 .ptr);  
428 0429 4  
429 0430 4 END  
430 0431 3 ELSE  
431 0432 3 cm_count = 1;  
432 0433 2  
433 0434 2  
434 0435 2 length = .ptr - nml$st_prmbuffer;  
435 0436 2 nml$addmsgprm (.bufdsc,  
436 0437 2 .msgsize,  
437 0438 2 .sem_list [pst$w_dataid],  
438 0439 2 nma$m_pty_cmu OR .cm_count,  
439 0440 2 .length.
```

NMLSLISPRM
V04-000

NML special parameter handling routines
NMLSI ISNODEID Get host node id

NHLSLISNODEID Get host node 18

K 1
16-Sep-1984 00:16:56 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:50:09 [NML.SRC]NML!SPRM.B32;1

Page 13
(5)

```
445     0441 2             nmlSt_prmbuffer);  
446     0442 3  
447     0443 2 RETURN nmlS_sts_suc  
448     0444 1 END;
```

! End of NMLSLISNODEID

.PSECT SPLIT,NOWRT,NOEXE,2

00000006. 00008 P.AAB: .LONG 6
00000000. 0000C .ADDRESS TMPBUFFER

.PSECT SOUNDS, NO EXE, 2

00148 TMPBUFFER:

.BLKB 6

TMPDSC= P.AAB

.PSECT SCODES,NOWRT,2

		01FC	00000	.ENTRY	NML\$LI\$NODEID, Save R2,R3,R4,R5,R6,R7,R8
58	00000000'	00	9E 00002	MOVAB	NML\$T PRMBUFFÉR, R8
5E		10	C2 00009	SUBL2	#16, SP
		7E	D4 0000C	CLRL	FLDADR
			5E DD 0000E	PUSHL	SP
56	08	AE	9F 00010	PUSHAB	FLDSIZE
7E	04	AC	DD 00013	MOVL	SEM_LIST, R6
	66	3C	00017	MOVZWL	(R6), -(SP)
00000000G	56	10	AC DD 0001A	PUSHL	DATDSC
00	00	04	FB 0001D	CALLS	#4, NMASSEARCHFLD
04		50	E8 00024	BLBS	R0, 18
50		0C	CE 00027	MNEGL	#12, R0
			04 0002A	RET	
53		68	9E 0002B	18:	MOVAB
06		00	EF 0002E	EXTZV	NML\$T PRMBUFFER, PTR
50	00	BE	B0 00034	MOVW	#0, #8, 3(R6), CM_COUNT
83		02	90 00038	MOVBL	FLDADR, HOSTADR
03	00000000G	00	91 0003B	CMPB	#2, (PTR)+
		15	1A 00042	BGTRU	NML\$GB_NCP_VERSION, #3
06		02	EF 00044	EXTZV	28
06		0A	ED 0004D	CMPZV	#2, #6, NML\$GW_PERM EXEC_ADDR+1, R1
		05	12 00052	BNEQ	#10, #6, HOSTADR, RT
50	FC00	8F	AA 00054	BICW2	28
83		50	B0 00059	MOVW	#64512, HOSTADR
02		57	D1 0005C	CMPL	HOSTADR, (PTR)+
		35	12 0005F	BNEQ	CM_COUNT, #2
		08	AE 9F 00061	48	
00000000G	00000000'	00	9F 00064	PUSHAB	RESLEN
7E		50	3C 0006A	PUSHAB	TMPDSC
00		03	FB 0006D	MOVZWL	HOSTADR, -(SP)
AE	08	AE	B0 00074	CALLS	#3, NML\$GETNODNAME
AE	0148	C8	9E 00079	MOVW	RE\$LEN, NAMDSC
50	0C	AE	3C 0007F	MOVAB	TMPBUFFER, NAMDSC+4
		0E	13 00083	MOVZWL	NAMDSC, R0
83	40	8F	90 00085	BEQL	38
				MOVBL	#64, (PTR)+

NMLSLISPRM
V04-000

NML special parameter handling routines
NMLSLISNODEID Get host node id

L 3
16-Sep-1984 00:16:56
14-Sep-1984 12:50:09
VAX-11 Bliss-32 V4.0-742
[NML.SRC]NMLLISPRM.B32;1

Page 14
(5)

63	10	83	50	90	00089	MOV B	R0, (PTR)+	: 0426
		BE	50	28	0008C	MOVC3	R0, @NAMDSC+4, (PTR)	: 0429
			03	11	00091	BRB	4\$: 0433
		57	01	00	00093	38:	MOVL #1, CM COUNT	: 0432
50	50	50	68	9E	00096	48:	MOVAB NML\$T PRMBUFFER, R0	: 0435
	53		50	C3	00099	SUBL3	R0 PTR LENGTH	
		0101	8F	BB	0009D	PUSHR #^H<R0,R8>		
7E	57	0000000C0	8F	C9	000A1	BISL3	#192, CM COUNT, -(SP)	: 0440
	7E		66	3C	000A9	MOVZWL (R6), -(SP)	: 0439	
	7E	08	AC	7D	000AC	MOVO BUFDSC, -(SP)	: 0438	
00000000G	00		06	FB	000B0	CALLS #6, NMLLISPRM.B32;1	: 0436	
	50		01	00	000B7	MOVL #1, R0		
			04	000BA		RET		: 0443
								: 0444

: Routine Size: 187 bytes, Routine Base: \$CODE\$ + 00B2

0445 1 ZSBTTL 'NMLSLISPARAM Get parameter'
0446 1 GLOBAL ROUTINE NMLSLISPARAM (SEM_LIST, BUFDSC, MSGSIZE, DATDSC)=
0447 1 !++
0448 1 FUNCTIONAL DESCRIPTION:
0449 1 This routine returns a parameter.
0450 1 FORMAL PARAMETERS:
0451 1 SEM LIST Parameter semantic table entry address.
0452 1 BUFDSC Output message buffer descriptor address.
0453 1 MSGSIZE Address of current output message size.
0454 1 DATDSC QIO buffer descriptor address.
0455 1 IMPLICIT INPUTS:
0456 1 It is assumed that the permanent data base file is already open.
0457 1 IMPLICIT OUTPUTS:
0458 1 The output message buffer contains the coded multiple version number.
0459 1 ROUTINE VALUE:
0460 1 COMPLETION CODES:
0461 1 Always returns success (NML\$STS_SUC).
0462 1 SIDE EFFECTS:
0463 1 NONE
0464 1 --
0465 1 BEGIN
0466 1 MAP
0467 1 SEM_LIST : REF BBLOCK;
0468 1 LOCAL
0469 1 DATATYPE : BBLOCK [1], ! NICE parameter data type.
0470 1 FLDADR,
0471 1 FLDSIZE;
0472 1 FLDADR = 0;
0473 1 IF NMASSEARCHFLD (.DATDSC,
0474 1 .SEM_LIST [PSTS_W_DATAID],
0475 1 FLDSIZE,
0476 1 FLDADR)
0477 1 THEN
0478 1 BEGIN
0479 1 DATATYPE = .SEM_LIST [PSTS_B_DATATYPE];
0480 1 ! If the parameter is not an ASCII or hex image field, the length
0481 1 goes in the datatype byte. Add it here.
0482 1 !
0483 1
0484 1
0485 1
0486 1
0487 1
0488 1
0489 1
0490 1
0491 1
0492 1
0493 1
0494 1
0495 1
0496 1
0497 1
0498 1
0499 1
0500 1
0501 1

```

507 0502 3 IF (NOT .DATATYPE [NMASV_PTY ASC]) AND
508 0503 3 (.DATATYPE [NMASV_PTY-TYP] NEQ NMASC_PTY_HI) THEN
509 0504 3 DATATYPE = .DATATYPE OR .FLDSIZE;
510 0505 3 NML$ADDMSGPRM (.BUFDSC,
511 0506 3 .MSGSIZE
512 0507 3 .SEM_LIST [PSTS_W_DATAID],
513 0508 3 .DATATYPE,
514 0509 3 .FLDSIZE,
515 0510 3 .FLDADR);
516 0511 2 END;
517 0512 2 RETURN NMLS_STS_SUC
518 0513 2
519 0514 1 END;

```

! End of NMLSLISPARAM

			0004 00000	.ENTRY NML\$LISPARAM, Save R2	: 0446
		5E	04 C2 00002	SUBL2 #4, SP	
			7E D4 00005	CLRL FLDADR	: 0489
		08	SE DD 00007	PUSHL SP	: 0491
		52	04 AE 9F 00009	PUSHAB FLDSIZE	
		7E	62 3C 00010	MOVL SEM_LIST, R2	: 0492
			AC DD 00013	MOVZWL (R2), -(SP)	
		10	04 04 FB 00016	PUSHL DATDSC	: 0491
		00000000G	50 E9 0001D	CALLS #4, NMASSEARCHFLD	
		29	50 A2 90 00020	BLBC R0, 2\$	
		50	06 E0 00024	MOVB 3(R2), DATATYPE	: 0497
		50	00 ED 00028	BBS #6, DATATYPE, 1\$: 0502
20	50	0F	04 13 0002D	CMPZV #0, #15, DATATYPE, #32	: 0503
		50	04 AE 88 0002F	BEQL 1\$	
			6E DD 00033	BISB2 FLDSIZE, DATATYPE	: 0504
		08	AE DD 00035	PUSHL FLDADR	: 0510
		7E	50 9A 00038	PUSHL FLDSIZE	: 0509
		7E	62 3C 0003B	MOVZBL DATATYPE, -(SP)	: 0508
		7E	08 AC 7D 0003E	MOVZWL (R2), -(SP)	: 0507
		00000000G	06 FB 00042	MOVL MOVDSC, -(SP)	: 0505
		50	01 D0 00049	CALLS #6, NMLLISPRM	
			04 0004C	MOVL #1, R0	: 0513
				RET	: 0514

; Routine Size: 77 bytes, Routine Base: SCODE\$ + 0160

```
521 0515 1 XSBTTL 'NML$LI$PASSWORD Get parameter'  
522 0516 1 GLOBAL ROUTINE NML$LI$PASSWORD (SEM_LIST, BUFDSC, MSGSIZE, DATDSC)=  
523 0517 1  
524 0518 1 ++  
525 0519 1 FUNCTIONAL DESCRIPTION:  
526 0520 1  
527 0521 1 This routine adds a password parameter to the output message if  
528 0522 1 the user has the BYPASS privilege.  
529 0523 1  
530 0524 1 FORMAL PARAMETERS:  
531 0525 1  
532 0526 1 SEM_LIST Parameter semantic table entry address.  
533 0527 1 BUFDSC Output message buffer descriptor address.  
534 0528 1 MSGSIZE Address of current output message size.  
535 0529 1 DATDSC Address of data buffer descriptor.  
536 0530 1  
537 0531 1 IMPLICIT INPUTS:  
538 0532 1  
539 0533 1 It is assumed that the permanent data base file is already open.  
540 0534 1  
541 0535 1 IMPLICIT OUTPUTS:  
542 0536 1  
543 0537 1 NONE  
544 0538 1  
545 0539 1 ROUTINE VALUE:  
546 0540 1 COMPLETION CODES:  
547 0541 1  
548 0542 1 Always returns success (NMLS_STS_SUC).  
549 0543 1  
550 0544 1 SIDE EFFECTS:  
551 0545 1  
552 0546 1 NONE  
553 0547 1 --  
554 0548 1  
555 0549 1  
556 0550 2 BEGIN  
557 0551 2  
558 0552 2 MAP  
559 0553 2 SEM_LIST : REF BBLOCK;  
560 0554 2  
561 0555 2 BIND  
562 0556 2 STRDSC = $ASCID ('no access rights') : DESCRIPTOR;  
563 0557 2  
564 0558 2 LOCAL  
565 0559 2 FLDADR,  
566 0560 2 FLDSIZE;  
567 0561 2  
568 0562 2 IF NOT .NML$GQ_PROPVRMSK [PRV$V_BYPASS]  
569 0563 2 THEN  
570 0564 2 BEGIN  
571 0565 2  
572 0566 2 User does not have BYPASS privilege so return string to indicate that  
573 0567 2 a password is set if one is found.  
574 0568 2  
575 0569 2  
576 0570 2 FLDADR = 0;  
577 0571 2 IF NMASSEARCHFLD (.DATDSC,
```

```

578      0572 3          .SEM_LIST [PSTSW_DATAID],
579      0573 3          FLDSIZE,
580      0574 3          FLDADR)
581      0575 3          THEN
582      0576 4          BEGIN
583      0577 4
584      0578 4          NML$ADDMSGPRM (.BUFDSC,
585      0579 4          .MSGSIZE,
586      0580 4          .SEM_LIST [PSTSW_DATAID],
587      0581 4          .SEM_LIST [PSTSB-DATATYPE],
588      0582 4          .STRDSC [DSCSW_LENGTH],
589      0583 4          .STRDSC [DSCSA_POINTER]);
590      0584 4
591      0585 4          RETURN NMLS_STS_SUC
592      0586 4
593      0587 3          END;
594      0588 2          END;
595      0589 2
596      0590 2          ! Call the normal parameter routine.
597      0591 2
598      0592 2          NML$LISPARAM (.SEM_LIST,
599      0593 2          .BUFDSC,
600      0594 2          .MSGSIZE,
601      0595 2          .DATDSC);
602      0596 2
603      0597 2          RETURN NMLS_STS_SUC
604      0598 1          END;

```

! End of NMLSLISPASSWORD

74 68 67 69 72 20 73 73 65 63 63 61 20 6F 6E 00010 P.AAD: .ASCII \no access rights\
73 0001F
00000010, 00020 P.AAC: .LONG 16
00000000, 00024 .ADDRESS P.AAD

STRDSC= P-AAC

PSECT SCODES, NOVRT, 2

			0004	00000	.ENTRY	NMLSLISPASSWORD, Save R2
3C 00000000G	5E 00		08 C2	00002	SUBL2	#8, SP
			05 E0	00005	BBS	#5, NMLSGQ_PROPRVMSK+3, 18
			6E D4	0000D	CLRL	FLDADR
			5E DD	0000F	PUSHL	SP
		08	AE 9F	00011	PUSHAB	FLDSIZE
52	04		AC DD	00014	MOVL	SEM LIST, R2
7E			62 3C	00018	MOVZWL	(R2), -(SP)
		10	AC DD	0001B	PUSHL	DATDSC
00000000G	00		04 FB	0001E	CALLS	#4, NMASSEARCHFLD
	21		50 E9	00025	BLBC	R0, 18
		00000000	00 DD	00028	PUSHL	STRDSC+4
	7E	00000000	00 3C	0002E	MOVZWL	STRDSC, -(SP)
	7E	03	A2 9A	00035	MOVZBL	3(R2), -(SP)
	7E		62 3C	00039	MOVZWL	(R2), -(SP)

NMLS I SPRM
V04-000

NML special parameter handling routines

NMLSLISPASSWORD Get parameter

D 4
16-Sep-1984 00:16:56 VAX-11 BLiss-32 V4.0-742
14-Sep-1984 12:50:09 [NML.SRC]NMLLISPRM.B32;1

Page 19
(7)

; Routine Size: 90 bytes, Routine Base: SCODES + 01BA

NML
V04

```
606      0599 1 XSBTTL 'NML$LISPSET List password set'
607      0600 1 GLOBAL ROUTINE NML$LISPSET (SEM_LIST, BUFDSC, MSGSIZE, DATDSC)=
608
609      0601 1 ++
610      0602 1 :++ FUNCTIONAL DESCRIPTION:
611      0603 1 :+
612      0604 1 :+ This routine is called while processing a LIST X25-S or X29-S DEST
613      0605 1 :+ command. If a password is set, it adds a password set indicator to
614      0606 1 :+ the NICÉ response message.
615      0607 1 :+
616      0608 1 :+ FORMAL PARAMETERS:
617      0609 1 :+
618      0610 1 :+ SEM_LIST      Parameter semantic table entry address.
619      0611 1 :+ BUFDSC        Output message buffer descriptor address.
620      0612 1 :+ MSGSIZE       Address of current output message size.
621      0613 1 :+ DATDSC        Address of data buffer descriptor.
622      0614 1 :+
623      0615 1 :+ IMPLICIT INPUTS:
624      0616 1 :+
625      0617 1 :+ IMPLICIT OUTPUTS:
626      0618 1 :+
627      0619 1 :+ ROUTINE VALUE:
628      0620 1 :+ COMPLETION CODES:
629      0621 1 :+
630      0622 1 :+ SIDE EFFECTS:
631      0623 1 :+
632      0624 1 :+
633      0625 1 :--+
634      0626 1 :+
635      0627 2 BEGIN
636      0628 2 :+
637      0629 2 :+ MAP
638      0630 2 :+   SEM_LIST : REF BBLOCK;
639      0631 2 :+
640      0632 2 :+ LOCAL
641      0633 2 :+   FLDSIZE,
642      0634 2 :+   FLDADR;
643      0635 2 :+
644      0636 2 :+ IF NMASSEARCHFLD (.DATDSC,
645      0637 2 :+   .SEM_LIST [PST$W_DATAID],
646      0638 2 :+   FLDSIZE,
647      0639 2 :+   FLDADR) THEN
648      0640 3 :+ BEGIN
649      0641 3 :+   Add password to message with a value of 0. This indicates simply that
650      0642 3 :+   the password is defined, without actually returning the password.
651      0643 3 :+
652      0644 3 :+   NML$ADDMSGPRM (.BUFDSC,
653      0645 3 :+     .MSGSIZE,
654      0646 3 :+     .SEM_LIST [PST$W_DATAID],
655      0647 3 :+     1,
656      0648 3 :+     1,
657      0649 3 :+     UPLIT (0));
658      0650 3 :+
659      0651 2 :+ END;
660      0652 2 RETURN NMLS_STS_SUC
661      0653 1 END;
```

: end of NML\$LISPSET

```

.PSECT $SPLITS,NOWRT,NOEXE,2
00000000 00028 P.AAE: .LONG 0

.PSECT $CODES,NOWRT,2
: 0600
: 0636
: 0637
: 0636
: 0650
: 0645
: 0647
: 0645
: 0652
: 0653

      0000 0000
      08 C2 00002
      5E DD 00005
      AE 9F 00007
      BC 3C 0000A
      AC DD 0000E
      FB 00011
      E9 00018
      9F 0001B
      DD 00021
      DD 00023
      BC 3C 00025
      AC 7D 00029
      FB 0002D
      D0 00034 1$: 04 00037

      ENTRY NML$LISPWSET, Save nothing
      SUBL2 #8, SP
      PUSHL SP
      PUSHAB FLDSIZE
      MOVZWL @SEM_LIST, -(SP)
      PUSHL DATDSC
      CALLS #4, NMASSEARCHFLD
      BLBC R0, 1$
      PUSHAB P.AAE
      PUSHL #1
      PUSHL #1
      MOVZWL @SEM_LIST, -(SP)
      MOVQ BUFDSC, -(SP)
      CALLS #6, NML$ADDMSGPRM
      MOVL #1, R0
      RET

```

: Routine Size: 56 bytes, Routine Base: \$CODES + 0214

```

662      0654 1 ZSBTTL 'NMLSLISRANGE List range parameter'
663      0655 1 GLOBAL ROUTINE NMLSLISRANGE (SEM_LIST, BUFDSC, MSGSIZE, DATDSC)=
664      0656 1
665      0657 1 !++
666      0658 1 ! FUNCTIONAL DESCRIPTION:
667      0659 1
668      0660 1 This routine is called to list X25 and X29 Destination subaddresses
669      0661 1 and X25 DTE channels. The destination's subaddresses can be more
670      0662 1 than one range pair, in which case the field length in the permanent
671      0663 1 database is the number of range pairs times 4 (i.e. then length in
672      0664 1 bytes).
673      0665 1
674      0666 1 ! FORMAL PARAMETERS:
675      0667 1
676      0668 1 SEM_LIST      Parameter semantic table entry address.
677      0669 1 BUFDSC        Output message buffer descriptor address.
678      0670 1 MSGSIZE        Address of current output message size.
679      0671 1 DATDSC         Address of data buffer descriptor.
680      0672 1
681      0673 1 !--
682      0674 1
683      0675 2 BEGIN
684      0676 2
685      0677 2 MAP
686      0678 2     SEM_LIST : REF BBLOCK;
687      0679 2
688      0680 2 LOCAL
689      0681 2     FLDADR,
690      0682 2     FLDSIZE,
691      0683 2     CM_COUNT,
692      0684 2     LENGTH,
693      0685 2     PTR,
694      0686 2     RANGE_BEGIN,
695      0687 2     RANGE_END;
696      0688 2
697      0689 2     FLDADR = 0;
698      0690 2
699      0691 2 IF NMASSEARCHFLD (.DATDSC,
700      0692 2     ;SEM_LIST [PST$W_DATAID],
701      0693 2     ;FLDSIZE,
702      0694 2     ;FLDADR) THEN
703      0695 2 BEGIN
704      0696 2
705      0697 2     For as many range pairs as are set, add them to the NICE response message
706      0698 2     in the form: Parameter ID, Coded multiple data type, word data type,
707      0699 2     range begin, word data type, range end.
708      0700 2
709      0701 3 WHILE .FLDSIZE GTR 0 DO
710      0702 3     BEGIN
711      0703 4     PTR = NMLST PRMBUFFER;
712      0704 4     CM_COUNT = T;
713      0705 4
714      0706 4     CHSWCHAR A (2, PTR);
715      0707 4     PTR = CH$MOVE (2, (.FLDADR) <0,16>, .PTR);
716      0708 4
717      0709 4     ! If the range begin = range end, don't include range end.
718      0710 4

```

```

719 0711 4 IF (.FLDADR) <0,16> NEQ (.FLDADR) <16,32> THEN
720 0712 5 BEGIN
721 0713 5 CM COUNT = .CM_COUNT + 1;
722 0714 5 CH$WCHAR A {2,-PTR};
723 0715 5 PTR = CH$MOVE (2, (.FLDADR) <16,32>, .PTR);
724 0716 4 END;
725 0717 4
726 0718 4 LENGTH = .PTR - NML$T_PRMBUFFER;
727 0719 4 NML$ADDMSGPRM (.BUFDSC,
728 0720 4 .MSGSIZE,
729 0721 4 .SEM_LIST [PSTS$W_DATAID],
730 0722 4 .SEM_LIST [PSTS$B_DATATYPE] OR .CM_COUNT,
731 0723 4 .LENGTH,
732 0724 4 NML$T_PRMBUFFER);

733 0725 4
734 0726 4 ! Increment pointer and length to get next range pair in the
735 0727 4 permanent data base record.
736 0728 4
737 0729 4 FLDADDR = .FLDADDR + 4;
738 0730 4 FLDSIZE = .FLDSIZE - 4;
739 0731 3 END;
740 0732 2 END;
741 0733 2
742 0734 2 RETURN NML$_STS_SUC
743 0735 1 END;

```

! end of NML\$LISRANGE

				.ENTRY	NML\$LISRANGE, Save R2,R3,R4,R5,R6	: 0655
56	00000000:	007C 00000	00 9E 00002	MOVAB	NML\$T_PRMBUFFER, R6	: 0655
SE			04 C2 00009	SUBL2	#4, SP	: 0689
			7E D4 0000C	CLRL	FLDADR	: 0691
			5E DD 0000E	PUSHL	SP	: 0691
7E	08	AE 9F 00010	PUSHAB	FLDSIZE	: 0692	
	04	BC 3C 00013	MOVZWL	@SEM_LIST, -(SP)	: 0692	
	10	AC DD 00017	PUSHL	DATDSC	: 0691	
00000000G	00	04 FB 0001A	CALLS	#4, NMASSEARCHFLD	: 0691	
56	50	E9 00021	BLBC	R0, 3\$: 0722	
53	04	AC D0 00024	MOVL	SEM_LIST, R3	: 0701	
	04	AE D5 00028	TSTL	FLDSIZE	: 0701	
		18: 4D 15 0002B	BLEQ	3\$: 0703	
			66 9E 0002D	MOVAB	NML\$T_PRMBUFFER, PTR	: 0704
			01 D0 00030	MOVL	#1, CM_COUNT	: 0704
	52	02 90 00033	MOVB	#2, (PTR)+	: 0706	
	54	00 BE B0 00036	MOVW	@FLDADR, (PTR)+	: 0707	
	82	6E D0 0003A	MOVL	FLDADR, R0	: 0711	
	82	51 02 A0 9E 0003D	MOVAB	2(R0), R1	: 0711	
	51	6E D1 00041	CMPL	FLDADR, R1	: 0711	
		51 09 13 00044	BEQL	2\$: 0713	
			54 D6 00046	INCL	CM_COUNT	: 0713
	82	02 90 00048	MOVB	#2, (PTR)+	: 0714	
	82	02 A0 B0 0004B	MOVW	2(R0), (PTR)+	: 0715	
55	50	66 9F 0004F	MOVAB	NML\$T_PRMBUFFER, R0	: 0718	
	52	50 C3 00052	SUBL3	R0, PTR, LENGTH	: 0723	
		0060 8F BB 00056	PUSHR	#^MCR5,R6>	: 0723	

NML\$LISTPRM
V04-000

NML special parameter handling routines
NML\$LISTRANGE List range parameter

16-Sep-1984 00:16:56
14-Sep-1984 12:50:09
VAX-11 Bliss-32 V4.0-742
[NML.SRC]NML\$LISTPRM.B32;1

Page 24
(9)

NML
V04

7E	50	03	A3	9A	0005A	MOVZBL	3(R3), R0	: 0722
	50		54	C9	0005E	BISL3	CM COUNT, R0, -(SP)	: 0721
	7E	04	BC	3C	00062	MOVZWL	@SEM LIST, -(SP)	: 0719
	7E	08	AC	7D	00066	MOVQ	BUFDSC, -(SP)	: 0729
00000000G	00		06	FB	0006A	CALLS	#6. NML\$ADDMSGPRM	: 0730
	6E		04	CO	00071	ADDL2	#4. FLDADR	: 0701
	04	AE		C4	00074	SUBL2	#4. FLDSIZE	: 0734
			AE	11	00078	BRB	1S	: 0735
	50		01	D0	0007A 38:	MOVL	#1, R0	
			04	0007D		RET		

: Routine Size: 126 bytes. Routine Base: \$CODES + 024C

: R

```

745 0736 1 XSBTTL 'NMLSLISOWNER Get OWNER parameter'
746 0737 1 GLOBAL ROUTINE NMLSLISOWNER (SEM_LIST, BUFDSC, MSGSIZE, DATDSC)=
747 0738 1
748 0739 1 !++
749 0740 1 ! FUNCTIONAL DESCRIPTION:
750 0741 1 This routine adds the circuit parameter, OWNER, to the NICE
751 0742 1 response message. The owner parameter is saved as a bit value.
752 0743 1 If it's set, the executor owns the circuit. Check to see if
753 0744 1 it's set, and, if so, return the executor node ID.
754 0745 1
755 0746 1 ! FORMAL PARAMETERS:
756 0747 1
757 0748 1 SEM LIST Parameter semantic table entry address.
758 0749 1 BUFDSC Output message buffer descriptor address.
759 0750 1 MSGSIZE Address of current output message size.
760 0751 1 DATDSC QIO buffer descriptor address.
761 0752 1
762 0753 1 ! IMPLICIT INPUTS:
763 0754 1 It is assumed that the permanent data base file is already open.
764 0755 1
765 0756 1 ! IMPLICIT OUTPUTS:
766 0757 1 The output message buffer contains the coded multiple executor node
767 0758 1 address.
768 0759 1
769 0760 1 ! ROUTINE VALUE:
770 0761 1 ! COMPLETION CODES:
771 0762 1 Always returns success (NMLS_STS_SUC).
772 0763 1
773 0764 1 !--
774 0765 1
775 0766 2 BEGIN
776 0767 2
777 0768 2 MAP
778 0769 2 SEM_LIST : REF BBLOCK;
779 0770 2
780 0771 2 BIND EXECUTOR = UPLIT BYTE
781 0772 2 (NMASM PTY COD+1 NMASC ENT_NOD, ! Entity type = node
782 0773 2 2, WORD (0)): ! Node address = executor
783 0774 2
784 0775 2 LOCAL
785 0776 2   FLDADR,
786 0777 2   FLDSIZE:
787 0778 2
788 0779 2 FLDADR = 0:
789 0780 2 IF NMASSEARCHFLD (.DATDSC,
790 0781 2   :SEM_LIST [PSTS_W_DATAID],
791 0782 2   FLDSIZE,
792 0783 2   FLDADR) THEN
793 0784 2 BEGIN
794 0785 2   IF ..FLDADR THEN
795 0786 2     NML$ADDMSGPRM (.BUFDSC,
796 0787 2       :MSGSIZE,
797 0788 2       :SEM_LIST [PSTS_W_DATAID],
798 0789 2       :SEM_LIST [PSTS_B_DATATYPE] OR 2,
799 0790 2       $,
800 0791 2       EXECUTOR);
801 0792 2

```

NMLSLISPRM
V04-000

NML special parameter handling routines
NMLSLISOWNER Get OWNER parameter

K 4
16-Sep-1984 00:16:56
14-Sep-1984 12:50:09
VAX-11 BLiss-32 V4.0-742
[NML.SRC]NMLSLISPRM.B32;1

Page 26
(10)

: 802 0793 2 RETURN NMLS_STS_SUC
: 803 0794 1 END;

! End of NMLSLISOWNER

.PSECT SPLITS,NOWRT,NOEXE,2

02 00 81 0002C P.AAF: .BYTE -127. 0, 2
0000 0002F .WORD 0

EXECUTOR= P.AAF

.PSECT \$CODES,NOWRT,2

		0004 00000	.ENTRY NMLSLISOWNER, Save R2	0737
		04 C2 00002	SUBL2 #4, SP	0779
		7E D4 00005	CLRL FLDAADR	0780
		5E DD 00007	PUSHL SP	
		08 AE 9F 00009	PUSHAB FLDSIZE	
	52 04	AC D0 0000C	MOVL SEM LIST, R2	0781
	7E 10	62 3C 00010	MOVZWL (R2), -(SP)	
00000000G	00 00	AC DD 00013	PUSHL DATDSC	0780
	22	04 FB 00016	CALLS #4, NMASSEARCHFLD	
	1E	50 E9 0001D	BLBC R0, 1\$	
	00 00000000	BE E9 00020	BLBC AFLDADR, 1\$	0785
	00 00000000	00 9F 00024	PUSHAB EXECUTOR	0786
	50 03	05 DD 0002A	PUSHL #5	
7E	50 03	A2 9A 0002C	MOVZBL 3(R2), R0	0789
	50	02 C9 00030	BISL3 #2, R0, -(SP)	
	7E 08	62 3C 00034	MOVZWL (R2), -(SP)	0788
00000000G	00 08	AC 7D 00037	MOVO BUFDSC, -(SP)	0786
	50 00	06 FB 0003B	CALLS #6, NML\$ADDMSGPRM	
	01 00	01 D0 00042	MOVL #1, R0	0793
	04 00045	04 00045	RET	0794

: Routine Size: 70 bytes. Routine Base: \$CODES + 02CA

```

805 0795 1 ISBTTL 'NMLSDEFPARAM Add parameter'
806 0796 1 GLOBAL ROUTINE NMLSDEFPARAM (SEM_LIST, BUFSIZE, LENGTH, ADDR, RTNDSC)=
807 0797 1
808 0798 1 !++
809 0799 1 : FUNCTIONAL DESCRIPTION:
810 0800 1
811 0801 1 This routine adds a parameter to a permanent data base record.
812 0802 1
813 0803 1 : FORMAL PARAMETERS:
814 0804 1
815 0805 1 SEM_LIST Parameter semantic table entry address.
816 0806 1 BUFSIZE Permanent database record maximum size.
817 0807 1 LENGTH Length of parameter to insert in record.
818 0808 1 ADDR Address of parameter to insert in record.
819 0809 1 RTNDSC Permanent database record buffer descriptor address.
820 0810 1
821 0811 1 : IMPLICIT INPUTS:
822 0812 1
823 0813 1 It is assumed that the permanent data base file is already open.
824 0814 1
825 0815 1 : IMPLICIT OUTPUTS:
826 0816 1
827 0817 1 The parameter is added to the record.
828 0818 1
829 0819 1 : ROUTINE VALUE:
830 0820 1 : COMPLETION CODES:
831 0821 1
832 0822 1 Always returns success (NMLS_STS_SUC).
833 0823 1
834 0824 1 : SIDE EFFECTS:
835 0825 1
836 0826 1 None
837 0827 1
838 0828 1 --
839 0829 1
840 0830 2 BEGIN
841 0831 2
842 0832 2 MAP
843 0833 2 SEM_LIST : REF BBLOCK;
844 0834 2
845 0835 2 IF NOT NMASINSERTFLD (.BUFSIZE,
846 0836 2 .SEM_LIST [PSTS_W_DATAID],
847 0837 2 .LENGTH,
848 0838 2 .ADDR,
849 0839 2 .RTNDSC)
850 0840 2
851 0841 2 THEN
852 0842 2 BEGIN
853 0843 2
854 0844 2 Insert failed.
855 0845 2 NML$AB_MSGBLOCK [MSB$L_FLAGS] = MSBSM_MSG_FLD; ! Set message text flag
856 0846 2 NML$AB_MSGBLOCK [MSB$B_CODE] = NMASC_STS_MPR; ! Add error code
857 0847 2 NML$AB_MSGBLOCK [MSB$L_TEXT] = NMLS_RECVFOVF;
858 0848 2
859 0849 2 RETURN NMLS_STS_MPR
860 0850 2
861 0851 2 END;

```



```

867 0856 1 XSBTTL 'NMLSDEFLINLY Add line type parameter'
868 0857 1 GLOBAL ROUTINE NMLSDEFLINLY (SEM_LIST, BUFDSC, LENGTH, ADDR, RTNDSC)=
869 0858 1
870 0859 1 !++
871 0860 1 | FUNCTIONAL DESCRIPTION:
872 0861 1 |
873 0862 1 | This routine adds the line type parameter to the permanent data
874 0863 1 | base record if the value is valid.
875 0864 1 |
876 0865 1 | FORMAL PARAMETERS:
877 0866 1 |
878 0867 1 | SEM_LIST Parameter semantic table entry address.
879 0868 1 | BUFSIZE Permanent database record maximum size.
880 0869 1 | LENGTH Length of parameter to insert in record.
881 0870 1 | ADDR Address of parameter to insert in record.
882 0871 1 | RTNDSC Permanent database record buffer descriptor address.
883 0872 1 |
884 0873 1 | IMPLICIT INPUTS:
885 0874 1 |
886 0875 1 | It is assumed that the permanent data base file is already open.
887 0876 1 |
888 0877 1 | IMPLICIT OUTPUTS:
889 0878 1 |
890 0879 1 | The parameter is added to the record.
891 0880 1 |
892 0881 1 | ROUTINE VALUE:
893 0882 1 | COMPLETION CODES:
894 0883 1 |
895 0884 1 | Always returns success (NMLS_STS_SUC).
896 0885 1 |
897 0886 1 | SIDE EFFECTS:
898 0887 1 |
899 0888 1 | NONE
900 0889 1 |
901 0890 1 | !--
902 0891 1 |
903 0892 2 BEGIN
904 0893 2 |
905 0894 2 MAP
906 0895 2 | SEM_LIST : REF BBLOCK;
907 0896 2 |
908 0897 2 LOCAL
909 0898 2 | FLDADDR,
910 0899 2 | FLDSIZE,
911 0900 2 | STATUS;
912 0901 2 |
913 0902 2 IF (.ADDR)<0,8> EQL NMASC_LINTY_POI
914 0903 2 THEN BEGIN
915 0904 2 |
916 0905 2 | FLDSIZE = 0;
917 0906 2 | IF NMASSEARCHFLD (.RTNDSC,
918 0907 2 | NMASC_PCLI_TRI,
919 0908 2 | FLDSIZE,
920 0909 2 | FLDADDR)
921 0910 2 |
922 0911 2 THEN BEGIN
923 0912 2 |

```

```
924 0913 4 |  
925 0914 4 | Line has tributary address so it cannot have type=POINT.  
926 0915 4 |  
927 0916 4 |  
928 0917 4 | NML$AB_MSGBLOCK [MSBSL_FLAGS] = MSBSM_DET_FLD;  
929 0918 4 | NML$AB_MSGBLOCK [MSBSB_CODE] = NMASC_STS_PVA;  
930 0919 4 | NML$AB_MSGBLOCK [MSBSW_DETAIL] = NMASC_PCLI_LTY;  
931 0920 4 |  
932 0921 4 | RETURN NMLS_STS_PVA  
933 0922 4 |  
934 0923 3 |  
935 0924 2 | END;  
936 0925 2 |  
937 0926 2 | STATUS = NMLSDEFPARAM (.SEM_LIST,  
938 0927 2 | .BUFDSC,  
939 0928 2 | .LENGTH,  
940 0929 2 | .ADDR  
941 0930 2 | .RTNDSC);  
942 0931 2 |  
943 0932 2 | RETURN .STATUS  
944 0933 2 |  
945 0934 1 | END; ! End of NMLSDEFINLTY
```

NML
V04

S2	00000000G	00	0004	00000	.	ENTRY	NMLSDEFINLTY,	Save R2		0857	
SE		08	C2	00009		MOVAB	NMLSAB_MSGBLOCK,	R2			
	10	BC	95	0000C		SUBL2	#8,	SP			
		2B	12	0000F		TSTB	QADDR			0902	
	04	AE	D4	00011		BNEQ	1\$				
		5E	DD	00014		CLRL	FLDSIZE			0906	
	08	AE	9F	00016		PUSHL	SP			0907	
7E	0474	8F	3C	00019		PUSHAB	FLDSIZE				
	14	AC	DD	0001E		MOVZWL	#1140,	-(SP)			
00000000G	00	04	FB	00021		PUSHL	RTNDSC				
	11	50	E9	00028		CALLS	#4,	NMASSEARCHFLD			
	62	02	D0	0002B		BLBC	R0,	1\$			
04	A2	10	8E	0002E		MOVL	#2	NMLSAB_MSGBLOCK		0917	
08	A2	0458	8F	B0	00032	MNEG8	#16	NMLSAB_MSGBLOCK+4		0918	
	50	20	CE	00038		MOVW	#1112,	NMLSAB_MSGBLOCK+8		0919	
			04	0003B		MNEG8	#32,	R0		0921	
	7E	10	AC	7D	0003C	18:	RET				
	7E	08	AC	7D	00040		MOVO	ADDR,	-(SP)	0929	
		04	AC	DD	00044		MOVO	BUFDSC,	-(SP)	0927	
FF7C	CF	05	FB	00047		PUSHL	SEM_LIST			0926	
			04	0004C		CALLS	#5,	NMLSDEFPARAM		0934	
						RET					

; Routine Size: 77 bytes, Routine Base: \$CODE\$ + 0348

947 0935 1 XSBTTL 'NML\$DEFLINTRI Add line tributary address parameter'
948 0936 1 GLOBAL ROUTINE NML\$DEFLINTRI (SEM_LIST, BUFDSC, LENGTH, ADDR, RTNDSC)=
949 0937 1
950 0938 1 ++
951 0939 1 FUNCTIONAL DESCRIPTION:
952 0940 1
953 0941 1 This routine adds the line tributary address parameter to the
954 0942 1 permanent data base record if it is valid for this line.
955 0943 1
956 0944 1 FORMAL PARAMETERS:
957 0945 1
958 0946 1 SEM_LIST Parameter semantic table entry address.
959 0947 1 BUFSIZE Permanent database record maximum size.
960 0948 1 LENGTH Length of parameter to insert in record.
961 0949 1 ADDR Address of parameter to insert in record.
962 0950 1 RTNDSC Permanent database record buffer descriptor address.
963 0951 1
964 0952 1 IMPLICIT INPUTS:
965 0953 1
966 0954 1 It is assumed that the permanent data base file is already open.
967 0955 1
968 0956 1 IMPLICIT OUTPUTS:
969 0957 1
970 0958 1 The parameter is added to the record.
971 0959 1
972 0960 1 ROUTINE VALUE:
973 0961 1 COMPLETION CODES:
974 0962 1
975 0963 1 Always returns success (NMLS_STS_SUC).
976 0964 1
977 0965 1 SIDE EFFECTS:
978 0966 1
979 0967 1 NONE
980 0968 1
981 0969 1 --
982 0970 1
983 0971 2 BEGIN
984 0972 2
985 0973 2 MAP
986 0974 2 SEM_LIST : REF BBLOCK;
987 0975 2
988 0976 2 LOCAL
989 0977 2 FLDADR,
990 0978 2 FLDSIZE,
991 0979 2 STATUS;
992 0980 2
993 0981 2 FLDSIZE = 0;
994 0982 2 IF NMAS\$SEARCHFLD (.RTNDSC,
995 0983 2 NMASC_PCLI_LTY,
996 0984 2 FLDSIZE,
997 0985 2 FLDADR)
998 0986 2
999 0987 2 THEN BEGIN
1000 0988 2
1001 0989 2 IF .(FLDADR)<0,8> EQL NMASC_LINTY_POI
1002 0990 2 THEN BEGIN
1003 0991 2

```

: 1004 0992 4 : Line has type=POINT so no tributary address can be specified.
: 1005 0993 4 :
: 1006 0994 4 :
: 1007 0995 4 :
: 1008 0996 4 NMLSAB_MSGBLOCK [MSBSL_FLAGS] = MSBSM_DET_FLD;
: 1009 0997 4 NMLSAB_MSGBLOCK [MSBSB_CODE] = NMASC_STS_PNA;
: 1010 0998 4 NMLSAB_MSGBLOCK [MSBSW_DETAIL] = NMASC_PCLI_TRI;
: 1011 0999 4 :
: 1012 1000 6 RETURN NMLS_STS_PNA
: 1013 1001 4 :
: 1014 1002 3 END;
: 1015 1003 2 END;
: 1016 1004 2 :
: 1017 1005 2 STATUS = NMLSDEFPARAM (.SEM_LIST,
: 1018 1006 2 .BUFDSC,
: 1019 1007 2 .LENGTH,
: 1020 1008 2 .ADDR,
: 1021 1009 2 .RTNDSC);
: 1022 1010 2 :
: 1023 1011 2 RETURN .STATUS
: 1024 1012 2 :
: 1025 1013 1 END; ! End of NMLSDEFLINTRI

```

				.ENTRY	NMLSDEFLINTRI, Save R2	0936
	52 00000000G	00 0004 00000	00 9E 00002	MOVAB	NMLSAB_MSGBLOCK, R2	
	SE	08 C2 00009	08 AE D4 0000C	SUBL2	#8, SP	
		04	AE DD 0000F	CLRL	FLDSIZE	0981
		08	AE 9F 00011	PUSHL	SP	0982
	7E	0458	8F 3C 00014	PUSHAB	FLDSIZE	
		14	AC DD 00019	MOVZWL	#1112, -(SP)	
	00000000G	00	04 FB 0001C	PUSHL	RTNDSC	
		16	50 E9 00023	CALLS	#4, NMASSEARCHFLD	
			00 BE 95 00026	BLBC	R0, 1S	
			11 12 00029	TSTB	@FLDADR	0989
		62	02 D0 0002B	BNEQ	1S	
	04 A2	16	8E 0002E	MOVL	#2, NMLSAB_MSGBLOCK	0996
	08 A2	0474	8F B0 00032	MNEG	#22, NMLSAB_MSGBLOCK+4	0997
	50	2C	CE 00038	MOVW	#1140, NMLSAB_MSGBLOCK+8	0998
			04 0003B	MNEGL	#44, R0	1000
		7E	10 AC 7D 0003C	RET		
		7E	08 AC 7D 00040	MOVQ	ADDR, -(SP)	1008
		04	AC DD 00044	MOVQ	BUFDSC, -(SP)	1006
	FF2F CF	05	FB 00047	PUSHL	SEM_LIST	1005
			04 0004C	CALLS	#5, NMLSDEFPARAM	
				RET		1013

; Routine Size: 77 bytes, Routine Base: \$CODE\$ + 0395

1027 1014 1 XSBTTL 'NMLSDEF NODE ADDR Add node address parameter'
1028 1015 1 GLOBAL ROUTINE NMLSDEF_NODE_ADDR (SEM_LIST, BUFDSC, LENGTH, ADDR, RTNDSC)=
1029 1016 1
1030 1017 1 ++
1031 1018 1 FUNCTIONAL DESCRIPTION:
1032 1019 1 This routine checks the node address parameter to make sure
1033 1020 1 it does not already exists in the node permanent database. If it does
1034 1021 1 not, it adds the node address to the permanent data base record.
1035 1022 1 This routine is not used to check for duplicate node names because
1036 1023 1 the node database name key is declared as 'noduplicates', so RMS
1037 1024 1 will do this check for node names when the record is written to
1038 1025 1 the file.
1039 1026 1
1040 1027 1 FORMAL PARAMETERS:
1041 1028 1 SEM_LIST Parameter semantic table entry address.
1042 1029 1 BUFSIZE Permanent database record maximum size.
1043 1030 1 LENGTH Length of parameter to insert in record.
1044 1031 1 ADDR Address of parameter to insert in record.
1045 1032 1 RTNDSC Permanent database record buffer descriptor address.
1046 1033 1
1047 1034 1 IMPLICIT INPUTS:
1048 1035 1 It is assumed that the permanent data base file is already open.
1049 1036 1
1050 1037 1 IMPLICIT OUTPUTS:
1051 1038 1 The parameter is added to the record.
1052 1039 1
1053 1040 1 ROUTINE VALUE:
1054 1041 1 COMPLETION CODES:
1055 1042 1 Returns success (NMLS_STS_SUC) if the node address is successfully
1056 1043 1 added to the permanent database record.
1057 1044 1 Returns nmls_sts_pva if the new address is already defined in the
1058 1045 1 node permanent database.
1059 1046 1
1060 1047 1 SIDE EFFECTS:
1061 1048 1 NONE
1062 1049 1
1063 1050 1 --
1064 1051 1
1065 1052 2 BEGIN
1066 1053 2
1067 1054 2 MAP
1068 1055 2 sem_list : REF BBLOCK,
1069 1056 2 rtndsc : REF DESCRIPTOR;
1070 1057 2
1071 1058 2 LOCAL
1072 1059 2 status;
1073 1060 2
1074 1061 2
1075 1062 2 If there's another node in the permanent database with the new address,
1076 1063 2 return an error message to NCP.
1077 1064 2
1078 1065 2 IF nml_find_duplicate_node (.sem_list, .bufdsc,
1079 1066 2 .length, .addr,
1080 1067 2 .rtndsc) THEN
1081 1068 3 BEGIN
1082 1069 3 nmlfab_msblockquote [msb\$v_det_fld] = 1;
1083 1070 3 nmlfab_msblockquote [msb\$b_code] = nmc\$sts_pva;

```

: 1084    1071 3      nml$ab_msgblock [msb$w_detail] = .sem_list [pst$w_dataid];
: 1085    1072 3      RETURN nm[$_sts_pva
: 1086    1073 2      END;
: 1087    1074 2
: 1088    1075 2
: 1089    1076 2      ! The node address is unique. Add it to the node's permanent database record.
: 1090    1077 2
: 1091    1078 2      status = nml$defparam (.sem_list,
: 1092          .bufdsc,
: 1093          .length,
: 1094          .addr,
: 1095          .rndsc);
: 1096    1082 2
: 1097    1083 2
: 1098    1084 2      RETURN .status
: 1099    1085 2
: 1086 1      END;

```

! End of NMLSDEF_NODE_ADDR

			0004 00000	.ENTRY	NML\$DEF_NODE_ADDR, Save R2	: 1015
	52 00000000G	00 9E 00002		MOVAB	NML\$AB_MSGBLOCK, R2	
	7E 10	AC 7D 00009		MOVQ	ADDR, -(SP)	: 1066
	7E 08	AC 7D 0000D		MOVQ	BUFDSC, -(SP)	: 1065
		04 AC DD 00011		PUSHL	SEM_LIST	
	00000000V	00 05 FB 00014		CALLS	#5, NML_FIND_DUPLICATE_NODE	
	10	50 E9 0001B		BLBC	R0, 1S	
	62	02 88 0001E		BISB2	#2, NML\$AB_MSGBLOCK	: 1069
	04 A2	10 8E 00021		MNEG8	#16, NML\$AB_MSGBLOCK+4	: 1070
	08 A2	04 BC B0 00025		MOVW	ASEM_LIST, NML\$AB_MSGBLOCK+8	: 1071
	50	20 CE 0002A		MNEGL	#32, R0	: 1072
		04 0002D	1S:	RET		
	7E 10	AC 7D 0002E		MOVQ	ADDR, -(SP)	: 1081
	7E 08	AC 7D 00032		MOVQ	BUFDSC, -(SP)	: 1079
		04 AC DD 00036		PUSHL	SEM_LIST	: 1078
	FIFO CF	05 FB 00039		CALLS	#5, NMLSDEFPARAM	
		04 0003E		RET		: 1086

: Routine Size: 63 bytes, Routine Base: \$CODES + 03E2

```

1101 1087 1 ZSBTTL 'NMLSDEF_EXEC_ID Add executor name or address parameter'
1102 1088 1 GLOBAL ROUTINE NMLSDEF_EXEC_ID (SEM_LIST, BUFDSC, LENGTH, ADDR, RTNDSC)=
1103 1089 1
1104 1090 1 !++
1105 1091 1 ! FUNCTIONAL DESCRIPTION:
1106 1092 1 This routine is called when processing a DEFINE EXECUTOR command
1107 1093 1 to change the name or address of the executor node. It checks
1108 1094 1 the new name or address parameter to determine if it's already
1109 1095 1 assigned to some other node. If it is, this means the identity
1110 1096 1 of the executor is being changed. Delete the remote entry with
1111 1097 1 that name or address. The new name or address is added to the
1112 1098 1 executor node permanent database record. It is written back
1113 1099 1 to the file later.
1114 1100 1
1115 1101 1 ! FORMAL PARAMETERS:
1116 1102 1 SEM_LIST Parameter semantic table entry address.
1117 1103 1 BUFSIZE Permanent database record maximum size.
1118 1104 1 LENGTH Length of parameter to insert in record.
1119 1105 1 ADDR Address of parameter to insert in record.
1120 1106 1 RTNDSC Permanent database record buffer descriptor address.
1121 1107 1
1122 1108 1 ! IMPLICIT INPUTS:
1123 1109 1 It is assumed that the permanent data base file is already open.
1124 1110 1
1125 1111 1 ! IMPLICIT OUTPUTS:
1126 1112 1 The new executor name or address is added to the record.
1127 1113 1
1128 1114 1 ! ROUTINE VALUE:
1129 1115 1 ! COMPLETION CODES:
1130 1116 1 Returns success (NMLS_STS_SUC) if the node address is successfully
1131 1117 1 added to the permanent database record.
1132 1118 1
1133 1119 1 ! SIDE EFFECTS:
1134 1120 1 If the new executor name or address is already assigned to some
1135 1121 1 other node in the permanent database, that remote node is deleted from
1136 1122 1 the database.
1137 1123 1
1138 1124 1 !--
1139 1125 1
1140 1126 2 BEGIN
1141 1127 2
1142 1128 2 MAP
1143 1129 2     addr : REF BBLOCK [2].
1144 1130 2     sem_list : REF BBLOCK;
1145 1131 2
1146 1132 2 LOCAL
1147 1133 2     status,
1148 1134 2     temp;
1149 1135 2
1150 1136 2 IF nml_find_duplicate_node (.sem_list, .bufdsc,
1151 1137 2     .length, .addr,
1152 1138 2     .rtndsc) THEN
1153 1139 3 BEGIN
1154 1140 3
1155 1141 3     The executor node identity is being changed to that of a node that's
1156 1142 3     already in the database. Delete the remote entry for that node (there
1157 1143 3     are no parameters that it makes sense to carry over in this case)

```

```

: 1158 1144 3 ; so the executor can become that node.
: 1159 1145 3
: 1160 1146 3 nml$delete_node_rec (.sem_list [pst$w_dataid],      ! Database key
: 1161 1147 3 .length);                                ! Name or address dsc.
: 1162 1148 3 nml$ab_msblock [msb$v_msg_f[d] = 1;
: 1163 1149 3 nml$ab_msblock [msb$1_text] = nml$_recdelet;
: 1164 1150 2 END;
: 1165 1151 2
: 1166 1152 2 ; Put the RMS "current record" pointer back to the executor node's
: 1167 1153 2 entry.
: 1168 1154 2
: 1169 1155 2 *****TEMPORARY
: 1170 1156 2 nml$gw_perm_exec_addr = 0;
: 1171 1157 2 ****
: 1172 1158 2 nml$getexecadr (temp);
: 1173 1159 2
: 1174 1160 2 ; If the new executor address is 0, leave it that way. If the area number
: 1175 1161 2 of the address is 0, then default it to area 1 (this is for DEFINE EXEC
: 1176 1162 2 ADDRESS only) so the exec will have a valid area number in the database.
: 1177 1163 2
: 1178 1164 2 IF .sem_list [pst$w_dataid] EQL nma$e_pcno_add THEN
: 1179 1165 2 BEGIN
: 1180 1166 3 IF .addr [nma$v_addr] NEQ 0 AND
: 1181 1167 3 .addr [nma$v_area] EQL 0 THEN
: 1182 1168 3     addr [nma$v_area] = 1;
: 1183 1169 2 END;
: 1184 1170 2 status = nml$defparam (.sem_list,
: 1185 1171 2     .bufdsc,
: 1186 1172 2     .length,
: 1187 1173 2     .addr,
: 1188 1174 2     .rndsc);
: 1189 1175 2
: 1190 1176 2 IF .sem_list [pst$w_dataid] EQL nma$e_pcno_add THEN
: 1191 1177 2     nml$gw_perm_exec_addr = .(.addr)<0,16>
: 1192 1178 2 ELSE
: 1193 1179 2 BEGIN
: 1194 1180 3     CHSMOVE (.length, .addr, .nml$qq_perm_exec_name_dsc [1]);
: 1195 1181 3     nml$qq_perm_exec_name_dsc [0] = .length;
: 1196 1182 2 END;
: 1197 1183 2 RETURN .status
: 1198 1184 2
: 1199 1185 1 END:                                ! End of NMLSDEF_EXEC_ID

```

<pre> 57 00000000G 00 00FC 00000 SE 04 C2 00009 14 AC DD 0000C 52 10 AC DD 0000F 52 DD 00013 7E 08 AC 7D 00015 53 04 AC DD 00019 53 DD 0001D 00000000V 00 05 FB 0001F </pre>	<pre> .ENTRY NMLSDEF_EXEC_ID, Save R2,R3,R4,R5,R6,R7 MOVAB NML\$GW_PERM_EXEC_ADDR, R7 SUBL2 #4, SP PUSHL RTNDSC MOVL ADDR, R2 PUSHL R2 MOVO BUFDSC, -(SP) MOVL SEM_LIST, R3 PUSHL R3 CALLS #5, NML_FIND_DUPLICATE_NODE </pre>
---	---

: 1088
: 1138
: 1137
: 1136

		1F		50	E9 00026	BLBC	R0, 1\$		1
			0C	AC	9F 00029	PUSHAB	LENGTH		1146
		00000000G	7E	63	3C 0002C	MOVZWL	(R3), -(SP)		1
		00000000G	00	02	FB 0002F	CALLS	#2. NMLSDELETE_NODE_REC		1
		00000000G	00	04	88 00036	BISB2	#4. NMLSAB_MSGBLOCK		1148
		00000000G	00 00000000G	8F	D0 0003D	MOVL	#NMLS RECDELET, NMLSAB_MSGBLOCK+12		1149
				67	D4 00048	CLRL	NML\$GQ_PERM_EXEC_ADDR		1156
				5E	DD 0004A	PUSHL	SP		1158
		00000000G	00	01	FB 0004C	CALLS	#1. NMLSGETEXECADR		1
		01F6	BF	63	B1 00053	CMPW	(R3), #502		1164
				13	12 00058	BNEQ	2\$		1
		03FF	BF	62	B3 0005A	BITW	(R2), #1023		1166
				0C	13 0005F	BEQL	2\$		1
		FC	BF	01	A2 93 00061	BITB	1(R2), #252		1167
				05	12 00066	BNEQ	2\$		1
62	06		0A	01	F0 00068	INSV	#1, #10, #6, (R2)		1168
				14	AC DD 0006D	PUSHL	RTNDSC		1174
			7E	08	52 DD 00070	PUSHL	R2		1173
		FE72	CF	53	DD 00076	MOVO	BUFDSC, -(SP)		1171
			56	05	FB 00078	PUSHL	R3		1170
		01F6	8F	50	D0 0007D	CALLS	#5. NMLSDEFPARAM		1
				63	B1 00080	MOVL	R0, STATUS		1
				05	12 00085	CMPW	(R3), #502		1176
			67	62	3C 00087	BNEQ	3\$		1
				14	11 0008A	MOVZWL	(R2), NML\$GQ_PERM_EXEC_ADDR		1177
60		50 00000000G	00	00	D0 0008C	BRB	4\$		1
		00000000G	62	0C	AC 28 00093	3S:	NML\$GQ_PERM_EXEC_NAME_DSC+4, R0		1180
			00	0C	AC D0 00098	MOVC3	LENGTH, (R2), (R0)		1181
			50		56 D0 000A0	MOVL	LENGTH, NML\$GQ_PERM_EXEC_NAME_DSC		1183
					04 000A3	RET	STATUS, R0		1185

; Routine Size: 164 bytes, Routine Base: \$CODES + 0421

```

1201 1186 1 ISBTTL 'NML_FIND_DUPLICATE_NODE' Check perm db for node id'
1202 1203 1204 1205 1206 1207 1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 1220 1221 1222 1223 1224 1225 1226 1227 1228 1229 1230 1231 1232 1233 1234 1235 1236 1237 1238 1239 1240 1241 1242 1243 1244 1245 1246 1247 1248 1249 1250 1251 1252 1253 1254 1255 1256 1257 1187 1188 1189 1190 1191 1192 1193 1194 1195 1196 1197 1198 1199 1200 1201 1202 1203 1204 1205 1206 1207 1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 1220 1221 1222 1223 1224 1225 1226 1227 1228 1229 1230 1231 1232 1233 1234 1235 1236 1237 1238 1239 1240 1241 1242 1243 1244 1245 1246 1247 1248 1249 1250 1251 1252 1253 1254 1255 1256 1257 1258 1259 1260 1261 1262 1263 1264 1265 1266 1267 1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280 1281 1282 1283 1284 1285 1286 1287 1288 1289 1290 1291 1292 1293 1294 1295 1296 1297 1298 1299 1300 1301 1302 1303 1304 1305 1306 1307 1308 1309 1310 1311 1312 1313 1314 1315 1316 1317 1318 1319 1320 1321 1322 1323 1324 1325 1326 1327 1328 1329 1330 1331 1332 1333 1334 1335 1336 1337 1338 1339 1340 1341 1342 1343 1344 1345 1346 1347 1348 1349 1350 1351 1352 1353 1354 1355 1356 1357 1358 1359 1360 1361 1362 1363 1364 1365 1366 1367 1368 1369 1370 1371 1372 1373 1374 1375 1376 1377 1378 1379 1380 1381 1382 1383 1384 1385 1386 1387 1388 1389 1390 1391 1392 1393 1394 1395 1396 1397 1398 1399 1400 1401 1402 1403 1404 1405 1406 1407 1408 1409 1410 1411 1412 1413 1414 1415 1416 1417 1418 1419 1420 1421 1422 1423 1424 1425 1426 1427 1428 1429 1430 1431 1432 1433 1434 1435 1436 1437 1438 1439 1440 1441 1442 1443 1444 1445 1446 1447 1448 1449 1450 1451 1452 1453 1454 1455 1456 1457 1458 1459 1460 1461 1462 1463 1464 1465 1466 1467 1468 1469 1470 1471 1472 1473 1474 1475 1476 1477 1478 1479 1480 1481 1482 1483 1484 1485 1486 1487 1488 1489 1490 1491 1492 1493 1494 1495 1496 1497 1498 1499 1500 1501 1502 1503 1504 1505 1506 1507 1508 1509 1510 1511 1512 1513 1514 1515 1516 1517 1518 1519 1520 1521 1522 1523 1524 1525 1526 1527 1528 1529 1530 1531 1532 1533 1534 1535 1536 1537 1538 1539 1540 1541 1542 1543 1544 1545 1546 1547 1548 1549 1550 1551 1552 1553 1554 1555 1556 1557 1558 1559 1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570 1571 1572 1573 1574 1575 1576 1577 1578 1579 1580 1581 1582 1583 1584 1585 1586 1587 1588 1589 1589 1590 1591 1592 1593 1594 1595 1596 1597 1598 1599 1599 1600 1601 1602 1603 1604 1605 1606 1607 1608 1609 1609 1610 1611 1612 1613 1614 1615 1616 1617 1618 1619 1619 1620 1621 1622 1623 1624 1625 1626 1627 1628 1629 1629 1630 1631 1632 1633 1634 1635 1636 1637 1638 1639 1639 1640 1641 1642 1643 1644 1645 1646 1647 1648 1649 1649 1650 1651 1652 1653 1654 1655 1656 1657 1658 1659 1659 1660 1661 1662 1663 1664 1665 1666 1667 1668 1669 1669 1670 1671 1672 1673 1674 1675 1676 1677 1678 1679 1679 1680 1681 1682 1683 1684 1685 1686 1687 1688 1689 1689 1690 1691 1692 1693 1694 1695 1696 1697 1698 1699 1699 1700 1701 1702 1703 1704 1705 1706 1707 1708 1709 1709 1710 1711 1712 1713 1714 1715 1716 1717 1718 1719 1719 1720 1721 1722 1723 1724 1725 1726 1727 1728 1729 1729 1730 1731 1732 1733 1734 1735 1736 1737 1738 1739 1739 1740 1741 1742 1743 1744 1745 1746 1747 1748 1749 1749 1750 1751 1752 1753 1754 1755 1756 1757 1758 1759 1759 1760 1761 1762 1763 1764 1765 1766 1767 1768 1769 1769 1770 1771 1772 1773 1774 1775 1776 1777 1778 1779 1779 1780 1781 1782 1783 1784 1785 1786 1787 1788 1789 1789 1790 1791 1792 1793 1794 1795 1796 1797 1798 1799 1799 1800 1801 1802 1803 1804 1805 1806 1807 1808 1809 1809 1810 1811 1812 1813 1814 1815 1816 1817 1818 1819 1819 1820 1821 1822 1823 1824 1825 1826 1827 1828 1829 1829 1830 1831 1832 1833 1834 1835 1836 1837 1838 1839 1839 1840 1841 1842 1843 1844 1845 1846 1847 1848 1849 1849 1850 1851 1852 1853 1854 1855 1856 1857 1858 1859 1859 1860 1861 1862 1863 1864 1865 1866 1867 1868 1869 1869 1870 1871 1872 1873 1874 1875 1876 1877 1878 1879 1879 1880 1881 1882 1883 1884 1885 1886 1887 1888 1889 1889 1890 1891 1892 1893 1894 1895 1896 1897 1898 1899 1899 1900 1901 1902 1903 1904 1905 1906 1907 1908 1909 1909 1910 1911 1912 1913 1914 1915 1916 1917 1918 1919 1919 1920 1921 1922 1923 1924 1925 1926 1927 1928 1929 1929 1930 1931 1932 1933 1934 1935 1936 1937 1938 1939 1939 1940 1941 1942 1943 1944 1945 1946 1947 1948 1949 1949 1950 1951 1952 1953 1954 1955 1956 1957 1958 1959 1959 1960 1961 1962 1963 1964 1965 1966 1967 1968 1969 1969 1970 1971 1972 1973 1974 1975 1976 1977 1978 1979 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2019 2020 2021 2022 2023 2024 2025 2026 2027 2028 2029 2029 2030 2031 2032 2033 2034 2035 2036 2037 2038 2039 2039 2040 2041 2042 2043 2044 2045 2046 2047 2048 2049 2049 2050 2051 2052 2053 2054 2055 2056 2057 2058 2059 2059 2060 2061 2062 2063 2064 2065 2066 2067 2068 2069 2069 2070 2071 2072 2073 2074 2075 2076 2077 2078 2079 2079 2080 2081 2082 2083 2084 2085 2086 2087 2088 2089 2089 2090 2091 2092 2093 2094 2095 2096 2097 2098 2099 2099 2100 2101 2102 2103 2104 2105 2106 2107 2108 2109 2109 2110 2111 2112 2113 2114 2115 2116 2117 2118 2119 2119 2120 2121 2122 2123 2124 2125 2126 2127 2128 2129 2129 2130 2131 2132 2133 2134 2135 2136 2137 2138 2139 2139 2140 2141 2142 2143 2144 2145 2146 2147 2148 2149 2149 2150 2151 2152 2153 2154 2155 2156 2157 2158 2159 2159 2160 2161 2162 2163 2164 2165 2166 2167 2168 2169 2169 2170 2171 2172 2173 2174 2175 2176 2177 2178 2179 2179 2180 2181 2182 2183 2184 2185 2186 2187 2188 2189 2189 2190 2191 2192 2193 2194 2195 2196 2197 2198 2199 2199 2200 2201 2202 2203 2204 2205 2206 2207 2208 2209 2209 2210 2211 2212 2213 2214 2215 2216 2217 2218 2219 2219 2220 2221 2222 2223 2224 2225 2226 2227 2228 2229 2229 2230 2231 2232 2233 2234 2235 2236 2237 2238 2239 2239 2240 2241 2242 2243 2244 2245 2246 2247 2248 2249 2249 2250 2251 2252 2253 2254 2255 2256 2257 2258 2259 2259 2260 2261 2262 2263 2264 2265 2266 2267 2268 2269 2269 2270 2271 2272 2273 2274 2275 2276 2277 2278 2279 2279 2280 2281 2282 2283 2284 2285 2286 2287 2288 2289 2289 2290 2291 2292 2293 2294 2295 2296 2297 2298 2299 2299 2300 2301 2302 2303 2304 2305 2306 2307 2308 2309 2309 2310 2311 2312 2313 2314 2315 2316 2317 2318 2319 2319 2320 2321 2322 2323 2324 2325 2326 2327 2328 2329 2329 2330 2331 2332 2333 2334 2335 2336 2337 2338 2339 2339 2340 2341 2342 2343 2344 2345 2346 2347 2348 2349 2349 2350 2351 2352 2353 2354 2355 2356 2357 2358 2359 2359 2360 2361 2362 2363 2364 2365 2366 2367 2368 2369 2369 2370 2371 2372 2373 2374 2375 2376 2377 2378 2379 2379 2380 2381 2382 2383 2384 2385 2386 2387 2388 2389 2389 2390 2391 2392 2393 2394 2395 2396 2397 2398 2399 2399 2400 2401 2402 2403 2404 2405 2406 2407 2408 2409 2409 2410 2411 2412 2413 2414 2415 2416 2417 2418 2419 2419 2420 2421 2422 2423 2424 2425 2426 2427 2428 2429 2429 2430 2431 2432 2433 2434 2435 2436 2437 2438 2439 2439 2440 2441 2442 2443 2444 2445 2446 2447 2448 2449 2449 2450 2451 2452 2453 2454 2455 2456 2457 2458 2459 2459 2460 2461 2462 2463 2464 2465 2466 2467 2468 2469 2469 2470 2471 2472 2473 2474 2475 2476 2477 2478 2479 2479 2480 2481 2482 2483 2484 2485 2486 2487 2488 2489 2489 2490 2491 2492 2493 2494 2495 2496 2497 2498 2499 2499 2500 2501 2502 2503 2504 2505 2506 2507 2508 2509 2509 2510 2511 2512 2513 2514 25
```

003C 00000 NML FIND DUPLICATE NODE:

		10	BC	00031		QADDR		
		04	4F	13 00033		2\$		
04	AE	04	BC	3C 00035	1S:	MOVZWL	ASEM_LIST, KEY	1254
			5E	DD 0003A		PUSHL	SP	1255
		0C	AE	9F 0003C		PUSHAB	DUP DSC	
		00000000.	00	9F 0003F		PUSHAB	NML\$Q_PRMDSC	
		0C	AC	9F 00045		PUSHAB	LENGTH	
		14	AE	9F 00048		PUSHAB	KEY	
00000000G	00		7E	D4 0004B		CLRL	-(SP)	
			06	FB 0004D		CALLS	#6, NML\$READRECORD	
		54	50	DO 00054		MOVL	RO, STATUS	
		2D	54	E9 00057		BLBC	STATUS, 3\$	1261
04	A5	65	40	9A 0005A		MOVZBL	#64, NML\$Q_ENTBFDSC	1268
		C0	A5	9E 0005E		MOVAB	NML\$T_ENTBUFFER, NML\$Q_ENTBFDSC+4	1269
			55	DD 00063		PUSHL	R5	1273
			55	DD 00065		PUSHL	R5	1270
			08	AE DD 00067		PUSHL	NODE_TYPE	1271
			14	AE 9F 0006A		PUSHAB	DUP_DSC	1270
00000000G	00		04	FB 0006D		CALLS	#4, NML\$GETREOWNER	
00000000G	00		10	DO 00074		MOVL	#16, NML\$AB_MSGBLOCK	1274
00000000G	00		65	9E 0007B		MOVAB	NML\$Q_ENTBFDSC, NML\$AB_MSGBLOCK+20	1275
			03	11 00082		BRB	3\$	1249
		54	10	CE 00084	2\$:	MNEGL	#16, STATUS	1279
		50	54	DO 00087	3\$:	MOVL	STATUS, RO	1280
				04 0008A		RET		1281

: Routine Size: 139 bytes. Routine Base: \$CODE\$ + 04C5

```
1298 1 ZSBTTL 'NMLSDEFNODNL| Add loop node line parameter'
1299 1 GLOBAL ROUTINE NMLSDEFNODNL| (SEM_LIST, BUFDSC, LENGTH, ADDR, RTNDSC)=
1300 1 ++
1301 1 : FUNCTIONAL DESCRIPTION:
1302 1 :
1303 1 This routine adds the loop node line parameter to the permanent
1304 1 data base record if this is a loop node and the circuit id is
1305 1 unique (i.e. there is no other loop node set up on the circuit).
1306 1 :
1307 1 : FORMAL PARAMETERS:
1308 1 :
1309 1293 1 :
1310 1294 1 SEM_LIST Parameter semantic table entry address.
1311 1295 1 BUFSIZE Permanent database record maximum size.
1312 1296 1 LENGTH Length of parameter to insert in record.
1313 1297 1 ADDR Address of parameter to insert in record.
1314 1298 1 RTNDSC Permanent database record buffer descriptor address.
1315 1299 1 :
1300 1300 1 : IMPLICIT INPUTS:
1301 1301 1 It is assumed that the permanent data base file is already open.
1302 1302 1 :
1303 1303 1 : IMPLICIT OUTPUTS:
1304 1304 1 The parameter is added to the record.
1305 1305 1 :
1306 1306 1 : ROUTINE VALUE:
1307 1307 1 : COMPLETION CODES:
1308 1308 1 Always returns success (NMLS_STS_SUC).
1309 1309 1 :
1310 1310 1 : SIDE EFFECTS:
1311 1311 1 NONE
1312 1312 1 :
1313 1313 1 !--
1314 1314 2 BEGIN
1315 1315 2 :
1316 1316 2 MAP
1317 1317 2 sem_list : REF BBLOCK;
1318 1318 2 :
1319 1319 2 LOCAL
1320 1320 2 fldadr,
1321 1321 2 fldsize,
1322 1322 2 circuit_dsc: VECTOR [2] ! Circuit already in node record (if any)
1323 1323 2 node_rec_buf: BBLOCK [nm[$k_recflen], ! Buffer for node data
1324 1324 2 node_rec_dsc: VECTOR [2]; ! Descriptor of node record buffer.
1325 1325 2 node_rec_data: VECTOR [2]; ! Descriptor of data in node record buffer.
1326 1326 2 status;
1327 1327 2 :
1328 1328 2 fldadr = 0;
1329 1329 2 IF nma$searchfld (.rtndsc,
1330 1330 2 nma$c_pcno_add,
1331 1331 2 fldsize,
1332 1332 2 fldadr) THEN
1333 1333 2 :
1334 1334 3 BEGIN
1335 1335 3 :
1336 1336 3 Node has address so circuit is not allowed. Loopnodes have only one
1337 1337 3 parameter - a circuit ID.
1338 1338 3 :
```

```

1355      1339 3      nml$ab_msblock [msb$1_flags] = msb$1_det_fld;
1356      1340 3      nml$ab_msblock [msb$2_code] = nma$C_sts_pna;
1357      1341 3      nml$ab_msblock [msb$w_detail] = nma$C_pcno_nli;
1358      1342 2      RETURN nml$sts_pna
1359      1343 2      END;
1360      1344 2
1361      1345 2      circuit_dsc [0] = 0;
1362      1346 2      circuit_dsc [1] = 0;
1363      1347 2      status = nma$searchfld (.rndsc,
1364                      nma$C_pcno_nli,
1365                      circuit_dsc [0],
1366                      circuit_dsc [1]);
1367      1351 2
1368      1352 2      ! If the loop node is already set up on the circuit specified in the NICE
1369      1353 2      ! DEFINE command, I'm done. Otherwise, make sure the circuit isn't already
1370      1354 2      ! defined for some other loopnode.
1371      1355 2
1372      1356 2      IF NOT .status
1373      1357 2      OR (.status AND CHSNEQ (.circuit_dsc [0], .circuit_dsc [1],
1374                      .length, .addr)) THEN
1375      1359 2      BEGIN
1376      1360 2
1377      1361 2      ! Check to make sure there aren't any other loopnodes on the specified
1378      1362 2      ! circuit in the node database.
1379      1363 2
1380      1364 3      node_rec_dsc [0] = nml$k_recflen;
1381      1365 3      node_rec_dsc [1] = node_rec_buf;
1382      1366 3      node_rec_data [1] = node_rec_buf;
1383      1367 3      status = nml$read_loopnode (.length,
1384                                node_rec_dsc,           ! Address of circuit descriptor
1385                                node_rec_data);        ! I/O buffer descriptor
1386      1369 3                                node_rec_data);        ! Return node data descriptor
1387      1370 3      IF .status NEQ rms$eof THEN
1388      1371 4      BEGIN
1389      1372 4
1390      1373 4      ! Circuit name must be unique for loop node.
1391      1374 4
1392      1375 4      nml$g_entbfdsc [0] = nml$k_entbuflen;
1393      1376 4      nml$g_entbfdsc [1] = nml$g_entbuffer;
1394      1377 4      nml$g_getrecowner (node_rec_data,
1395                                nml$C_loopnode,
1396                                nml$g_entbfdsc,
1397                                nml$g_entbfdsc [0]);
1398      1381 4      nml$ab_msblock [msb$1_entity] = nml$g_entbfdsc; ! Add entity descriptor pointer
1399      1382 4      nml$ab_msblock [msb$1_flags] = msb$1_det_fld OR msb$1_entd_fld;
1400      1383 4      nml$ab_msblock [msb$2_code] = nma$C_sts_pva;
1401      1384 4      nml$ab_msblock [msb$w_detail] = nma$C_pcno_nli;
1402      1385 4      RETURN nml$sts_pva
1403      1386 3      END;
1404      1387 2
1405      1388 2
1406      1389 2      ! The circuit is not already DEFINEd for some other loopnode. Add it to
1407      1390 2      the node's permanent database record.
1408      1391 2
1409      1392 2      status = nml$defparam (.sem_list,
1410                      .bufdsc,
1411                      .length,
1412                      .addr,
1413                      .addr);

```

1412 1396 2 RETURN .status
1413 1397 2
1414 1398 1 END:

→ `findsc`):

! End of NML\$DEFNODNL!

NML\$LISPRM
V04-000

NML special parameter handling routines
NML\$DEFNODNL Add loop node line parameter

{ 6
16-Sep-1984 00:16:56 VAX-11 BLiss-32 V4.0-742
14-Sep-1984 12:50:09 [NML.SRC]NMLLISPRM.B32;1

Page 44
(17)

NML
V04

50	20	CE 000BE	MNEGL #32, R0	: 1385
		04 000C1	RET	
7E	10	AC 7D 000C2	3\$: MOVQ ADDR, -(SP)	: 1395
7E	08	AC 7D 000C6	MOVQ BUFDSC, -(SP)	: 1393
	04	AC DD 000CA	PUSHL SEM_LIST	: 1392
FCEE	CF	05 FB 000CD	CALLS #5, NML\$DEFPARAM	
	54	50 D0 000D2	MOVL R0, STATUS	
		04 0U0D5	RET	: 1398

: Routine Size: 214 bytes, Routine Base: \$CODES + 0550

1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472

1399 1 XSBTTL 'NML\$DEFOBJNUM Add object number parameter'
1400 1 GLOBAL ROUTINE NML\$DEFOBJNUM (SEM_LIST, BUFDSC, LENGTH, ADDR, RTNDSC)=
1401 1 !++
1402 1 | FUNCTIONAL DESCRIPTION:
1403 1 | This routine adds the object number parameter to the permanent
1404 1 | data base record if it is unique.
1405 1 | FORMAL PARAMETERS:
1406 1 |
1407 1 | SEM_LIST Parameter semantic table entry address.
1408 1 | BUFSIZE Permanent database record maximum size.
1409 1 | LENGTH Length of parameter to insert in record.
1410 1 | ADDR Address of parameter to insert in record.
1411 1 | RTNDSC Permanent database record buffer descriptor address.
1412 1 |
1413 1 | IMPLICIT INPUTS:
1414 1 | It is assumed that the permanent data base file is already open.
1415 1 |
1416 1 | IMPLICIT OUTPUTS:
1417 1 | The parameter is added to the record.
1418 1 |
1419 1 | ROUTINE VALUE:
1420 1 | COMPLETION CODES:
1421 1 |
1422 1 | Always returns success (NMLS_STS_SUC).
1423 1 |
1424 1 | SIDE EFFECTS:
1425 1 |
1426 1 | NONE
1427 1 |
1428 1 |
1429 1 |
1430 1 |
1431 1 |
1432 1 |
1433 1 |
1434 1 |
1435 2 | BEGIN
1436 2 |
1437 2 | MAP
1438 2 | SEM_LIST : REF BBLOCK;
1439 2 |
1440 2 | LOCAL
1441 2 | DUMDSC : DESCRIPTOR,
1442 2 | FLDADR,
1443 2 | FLDSIZE,
1444 2 | KEY : WORD,
1445 2 | STATUS;
1446 2 |
1447 2 | FLDADR = 0;
1448 2 | FLDSIZE = 0;
1449 2 | STATUS = NMASSEARCHFLD (.RTNDSC,
1450 2 | NMASC P'OB_NUM,
1451 2 | FLDSIZE,
1452 2 | FLDADR);
1453 2 |
1454 2 |
1455 2 | | If no object number is already defined or the object number is

```

: 1473      1456 2   ! changed by the command, and
: 1474      1457 2   ! the object number is not zero (duplicate objects numbered 0 are allowed),
: 1475      1458 2   ! make sure that the new object number is not already in the
: 1476      1459 2   ! permanent data base.
: 1477      1460 2
: 1478      1461 2
: 1479      1462 2
: 1480      1463 2
: 1481      1464 2
: 1482      1465 2
: 1483      1466 2
: 1484      1467 2
: 1485      1468 2
: 1486      1469 2
: 1487      1470 2
: 1488      1471 2
: 1489      1472 2
: 1490      1473 2
: 1491      1474 2
: 1492      1475 3
: 1493      1476 4
: 1494      1477 4
: 1495      1478 4 ! Object number is not unique.
: 1496      1479 4
: 1497      1480 4
: 1498      1481 4
: 1499      1482 4
: 1500      1483 4
: 1501      1484 4
: 1502      1485 4
: 1503      1486 3
: 1504      1487 2
: 1505      1488 2
: 1506      1489 2
: 1507      1490 2
: 1508      1491 2
: 1509      1492 2
: 1510      1493 2
: 1511      1494 2
: 1512      1495 2
: 1513      1496 2
: 1514      1497 1

      IF (.NOT .STATUS
           OR (.STATUS AND CH$NEQ (.FLDSIZE, .FLDADR, .LENGTH, .ADDR)))
           AND CH$NEQ (.LENGTH, UPLIT(0), .LENGTH, .ADDR))
      THEN
        BEGIN
          KEY = 0;
          IF NMASMATCHREC (NMASC_OPN_OBJ,
                            NMLSQ_PRMDSC,
                            KEY,
                            NMASC_PCOB_NUM,
                            .LENGTH,
                            .ADDR,
                            DUMDSC)
        THEN
          BEGIN
            ! Object number is not unique.
            NML$AB_MSGBLOCK [MSBSL_FLAGS] = MSBSM_DET_FLD;
            NML$AB_MSGBLOCK [MSBSB_CODE] = NMASC_STS_PVA;
            NML$AB_MSGBLOCK [MSBSW_DETAIL] = NMASC_PCOB_NUM;
            RETURN NMLS_STS_PVA
          END;
        END;
      STATUS = NMLSDEFPARAM (.SEM_LIST,
                             .BUFDSC,
                             .LENGTH,
                             .ADDR,
                             .RTNDSC);
      RETURN .STATUS
    END;
  ! End of NMLSDEFOBJNUM

```

.PSECT SPLITS,NOWRT,NOEXE,2

00000000	00031	.BLKB	3
	00034 P.AAG:	.LONG	0

.PSECT \$CODES,NOWRT,2

55 00000000G	00 003C 00000	.ENTRY NMLSDEFOBJNUM, Save R2,R3,R4,R5
5E	10 9E 00002	MOVAB NML\$AB_MSGBLOCK, R5
	10 C2 00009	SUBL2 #16, SP

: 1400

; Routine Size: 140 bytes, Routine Base: SCODES + 0626

```

1516 1498 1 %SBTTL 'NMLSPURPARAM Delete parameter'
1517 1499 1 GLOBAL ROUTINE NMLSPURPARAM (RTNDSC, SEM_LIST)=
1518 1500 1
1519 1501 1 !++
1520 1502 1 : FUNCTIONAL DESCRIPTION:
1521 1503 1
1522 1504 1 This routine removes a parameter from the permanent data base record.
1523 1505 1
1524 1506 1 : FORMAL PARAMETERS:
1525 1507 1
1526 1508 1 : SEM_LIST Parameter semantic table entry address.
1527 1509 1 : RTNDSC Record buffer descriptor address.
1528 1510 1
1529 1511 1 : IMPLICIT INPUTS:
1530 1512 1
1531 1513 1 It is assumed that the permanent data base file is already open.
1532 1514 1
1533 1515 1 : IMPLICIT OUTPUTS:
1534 1516 1
1535 1517 1 The parameter has been removed from the record.
1536 1518 1
1537 1519 1 : ROUTINE VALUE:
1538 1520 1 : COMPLETION CODES:
1539 1521 1
1540 1522 1 Always returns success (NMLS_STS_SUC).
1541 1523 1
1542 1524 1 : SIDE EFFECTS:
1543 1525 1
1544 1526 1 : NONE
1545 1527 1
1546 1528 1 --+
1547 1529 1
1548 1530 2 BEGIN
1549 1531 2
1550 1532 2 : MAP
1551 1533 2 : SEM_LIST : REF BBLOCK;
1552 1534 2
1553 1535 2 : NMASDELETEFLD (.RTNDSC,
1554 1536 2 : .SEM_LIST [PSTS_W_DATAID]);
1555 1537 2
1556 1538 2 RETURN NMLS_STS_SUC
1557 1539 2
1558 1540 1 END: ! End of NMLSPURPARAM

```

	7E	08	0000 0000	.ENTRY NMLSPURPARAM, Save nothing	: 1499
		04	BC 3C 00002	@SEM_LIST, -(SP)	: 1536
00000000G	00		AC DD 00006	PUSHL RTNDSC	: 1535
	50		02 FB 00009	CALLS #2, NMASDELETEFLD	: 1538
			01 D0 00010	MOVL #1, R0	: 1540
			04 00013	RET	

; Routine Size: 20 bytes, Routine Base: SCODES + 06B2

NML\$LISPRM
V04-000

NML special parameter handling routines
NML\$PURPARAM Delete parameter

H 6
14-Sep-1984 00:16:56 VAX-11 Bliss-32 V4.0-742
[NML.SRC]NMLLISPRM.B32;1

Page 49
(19)

```
1560 1541 1 ISBTTL 'NML$PURNNODNA Delete node name parameter'
1561 1542 1 GLOBAL ROUTINE NML$PURNNODNA (RTNDSC, SEM_LIST)=
1562 1543 1 ++
1563 1544 1 FUNCTIONAL DESCRIPTION:
1564 1545 1 This routine removes the node name parameter from the permanent
1565 1546 1 data base record if it is not required. It is required in the case
1566 1547 1 of a loop node.
1567 1548 1
1568 1549 1
1569 1550 1 FORMAL PARAMETERS:
1570 1551 1 RTNDSC Data buffer descriptor address.
1571 1552 1 SEM_LIST Parameter semantic table entry address.
1572 1553 1
1573 1554 1 IMPLICIT INPUTS:
1574 1555 1 It is assumed that the permanent data base file is already open.
1575 1556 1
1576 1557 1 IMPLICIT OUTPUTS:
1577 1558 1 NONE
1578 1559 1
1579 1560 1 ROUTINE VALUE:
1580 1561 1 COMPLETION CODES:
1581 1562 1 Error is returned if the parameter cannot be removed.
1582 1563 1
1583 1564 1 SIDE EFFECTS:
1584 1565 1 NONE
1585 1566 1
1586 1567 1 --
1587 1568 1
1588 1569 2 BEGIN
1589 1570 2
1590 1571 2 MAP
1591 1572 2 SEM_LIST : REF BBLOCK;
1592 1573 2
1593 1574 2 LOCAL
1594 1575 2 FLDADDR,
1595 1576 2 FLDSIZE;
1596 1577 2
1597 1578 2 FLDADDR = 0;
1598 1579 2 FLDSIZE = 0;
1599 1580 2 IF NMASSEARCHFLD (.RTNDSC,
1600 1581 2 NMASC_PNO_NLI,
1601 1582 2 FLDSIZE,
1602 1583 2 FLDADDR)
1603 1584 2 THEN
1604 1585 2 BEGIN
1605 1586 2
1606 1587 3 Node has circuit (is a loopnode) so name cannot be deleted.
1607 1588 3
1608 1589 3 NMLSAB_MSGBLOCK [MSBSL_FLAGS] = MSBSM_DET_FLD;
1609 1590 3 NMLSAB_MSGBLOCK [MSBSB_CODE] = NMASC_STS_PNA;
1610 1591 3 NMLSAB_MSGBLOCK [MSBSW_DETAIL] = NMASC_PNO_NNA;
1611 1592 3
1612 1593 3 RETURN NMLS_STS_PNA
1613 1594 3
1614 1595 3 END
1615 1596 2 ELSE
1616 1597 2 NMASDELETEFLD (.RTNDSC, .SEM_LIST [PSTS DATAID]);
```

NMLSLISPRM
V04-000

NML special parameter handling routines
NML\$PURNODNNA Delete node name parameter

J 6
16-Sep-1984 00:16:56 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:50:09 [NML.SRC]NMLLISPRM.B32;1

Page 51
(20)

1617 1598 2 RETURN NML\$_STS_SUC
1618 1599 2
1619 1600 2
1620 1601 1 END;

! End of NML\$PURNODNNA

52	00000000G	0004	00000	.	ENTRY	NML\$PURNODNNA,	Save R2			1542
5E		00	9E	00002	MOVAB	NML\$AB_MSGBLOCK,	R2			
		04	C2	00009	SUBL2	#4,	SP			
			7E	D4 0000C	CLRL	FLDADR				1578
		04	AE	D4 0000E	CLRL	FLDSIZE				1579
			5E	DD 00011	PUSHL	SP				1580
		08	AE	9F 00013	PUSHAB	FLDSIZE				
7E	01F5	8F	3C	00016	MOVZWL	#501, -(SP)				
		04	AC	DD 0001B	PUSHL	RTNDSC				
00000000G	00		04	FB 0001E	CALLS	#4, NMASSEARCHFLD				
	11		50	E9 00025	BLBC	R0, 1\$				
	62		02	D0 00028	MOVL	#2, NML\$AB_MSGBLOCK				1589
04	A2	16	8E	0002B	MNEG B	#22, NML\$AB_MSGBLOCK+4				1590
08	A2	01F4	8F	B0 0002F	MOVW	#500, NML\$AB_MSGBLOCK+8				1591
	50		2C	CE 00035	MNEGL	#44, R0				1593
			04	00038	RET					
7E	08	BC	3C	00039 1\$:	MOVZWL	@SEM_LIST, -(SP)				1597
	04	AC	DD	0003D	PUSHL	RTNDSC				
00000000G	00		02	FB 00040	CALLS	#2, NMASDELETEFLD				
	50		01	D0 00047	MOVL	#1, R0				1599
			04	0004A	RET					1601

: Routine Size: 75 bytes, Routine Base: \$CODE\$ + 06C6

NMLSLISPRM
V04-000

NML special parameter handling routines
NML\$PURNODNNA Delete node name parameter

K 6
16-Sep-1984 00:16:56
14-Sep-1984 12:50:09

VAX-11 Bliss-32 V4.0-742
[NML.SRC]NMLLISPRM.B32;1

Page 52
(21)

: 1622 1602 1 END
: 1623 1603 1
: 1624 1604 0 ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
SOWNS	334	NOVEC, WRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
SPLITS	56	NOVEC,NOWRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
SCODES	1809	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
-\$255\$DUA2B:[NML.OBJ]NMLLIB.L32;1	341	42	12	27	00:00.1
-\$255\$DUA2B:[SHRLIB]NMALIBRY.L32;1	887	21	2	47	00:00.2
-\$255\$DUA2B:[SYSLIB]STARLET.L32;1	9776	4	0	581	00:02.2

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:NMLLISPRM/OBJ=OBJ\$:NMLLISPRM MSRC\$:NMLLISPRM/UPDATE=(ENH\$:NMLLISPRM)

Size: 1809 code + 390 data bytes
Run Time: 00:34.6
Elapsed Time: 01:30.8
Lines/CPU Min: 2781
Lexemes/CPU-Min: 13283
Memory Used: 131 pages
Compilation Complete

0284 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY